DEVELOPMENT OF A DECISION SUPPORT SYSTEM USING NETWORK REPAIR LEVEL ANALYSIS FOR DETERMINING TEST PROGRAM SET ACQUISITION COSTS

THESIS

Thomas J. Girz, B.S.
GS-12, USAF

AFIT/GLM/LSM/88S-24

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

2

AFIT/GLM/LSM/88S-24

DEVELOPMENT OF A DECISION SUPPORT SYSTEM USING NETWORK REPAIR LEVEL ANALYSIS FOR DETERMINING TEST PROGRAM SET ACQUISITION COSTS

THESIS

Thomas J. Girz, B.S.
GS-12, USAF

AFIT/GLM/LSM/88S-24

Approved for public release; distribution unlimited

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GLM/LSM/88S-24

DEVELOPMENT OF A DECISION SUPPORT SYSTEM
USING NETWORK REPAIR LEVEL ANALYSIS
FOR DETERMINING TEST PROGRAM SET ACQUISITION COSTS

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Master of Science in Logistics Management

Thomas J. Girz, B.S.
GS-12 USAF

September 1988

Approved for public release; distribution unlimited

Acknowledgements

The purpose of this research was to develop a software system that would enhance as well as expand the usefulness of the Network Repair Level Analysis model.

Many people have helped me in researching and writing this thesis. I would like thank Jim Callahan for suggesting this thesis topic and for providing me with insight into the problem. I would also like to thank my thesis advisor, Charles Youther for his patience and belief in my ability to perform the required research. As well, Linda Pangborn was very helpful by reviewing the software programs and making suggestions regarding their application. A special thanks also goes to my good friend Kevin Fisher for sharing his dBase III Plus programming expertise with me during many hours of trouble-shooting these programs over the phone. Finally, a very special thanks to my parents [redacted] and [redacted] Girz for their love and encouragement and for instilling in me the appreciation of a good education. This thesis is in part an expansion of a term paper written by myself and Captain Calvin J. Romrell for LOGM 615, Decision Support Systems. That term paper formed the basis for the written portion of this thesis.

Thomas J. Girz

Table of Contents

Page

List of Figures

Page

List of Tables

List of Programs

Page

Trademark Acknowledgements

dBASE III Plus is a registered trademark of Ashton-Tate Company.

IBM is a registered trademark of International Business Machines Corporation.

MS-DOS is a registered trademark of Multimate International, an Ashton-Tate Company.

Turbo Pascal is a registered trademark of Borland International

VAX is a trademark of International Business Machines Corporation.

Zenith is a registered trademark of Zenith Electronics Corporation.

AFIT/GLM/LSM/88S-24

Abstract

The purpose of this ~~research~~ thesis was to determine the applicability of developing a software system to enhance as well as expand the usefulness of the Network Repair Level Analysis (NRLA) model. Maintenance planners and acquisition managers need new tools in order to effectively plan and manage support programs in the face of increasing budget cuts. This research provides an initial specific decision support system (DSS) that enhances the usefulness of the NRLA model by providing a user interface and a database management system (DBMS) which facilitates model manipulation. This initial specific DSS also expands the model's usefulness by providing an analysis technique that generates cost guidelines used in the acquisition of test program sets.

Management personnel from the Aeronautical Systems Division at Wright-Patterson Air Force Base were interviewed and consulted throughout the system development process. The DSS was developed using dBase III Plus(TM). This included writing applications programs that make up the user interface and the DBMS, and integrating them with a database and the analysis models.

This DSS can be installed and used on an IBM(R) compatible micro-computer with a 20 megabyte hard drive, a monitor, and a 120 column printer.

DEVELOPMENT OF A DECISION SUPPORT SYSTEM USING NETWORK REPAIR LEVEL ANALYSIS FOR DETERMINING TEST PROGRAM SET ACQUISITION COSTS

I. Introduction

General Background

Support planners and acquisition managers face increasing difficulty holding down the rising support costs for today's weapons systems. They need new aids in making time critical, economically efficient, repair policy decisions. Economical maintenance and support concepts must be identified during weapon system acquisition and evaluated throughout the development process. The total life cycle cost of a weapon system depends on the development of system concepts that are economical to operate and support (12:3; 15:3). Early integrated logistics support (ILS) activity should focus on designing desirable support characteristics into weapon systems. This includes determining support requirements and evaluating alternative support concepts and techniques in order to minimize support cost risks (11:5). The repair level must be selected carefully since it can effect operational capabilities and cost.

An evaluation method called repair level analysis (RLA) helps in making this selection by determining if a weapon system's component is worth repairing and, if so, at what location (3:1; 15:2).

Repair Level Analysis. RLA is an iterative decision-making process conducted throughout all phases of a system's life cycle (16:2; 15:4). A primary goal of RLA is to make effective and economical repair level decisions which consider equipment design, operations, and logistics support characteristics. Through the RLA process, candidate items are analyzed and evaluated for discard at failure, intermediate level (I-level) repair, depot level (D-level) repair, or combinations thereof based on the equipment failure mode (16:1). RLA will establish the least cost repair or discard decision alternative and should be used to influence the system design of support resources in that direction.

Besides influencing the support system's design, the RLA decision process provides the basis for maintenance planning. The analysis indicates the repair location, extent of maintenance permitted, and the resources required to support the maintenance repair process. This information can be time critical in assessing integrated logistics support requirements and will, in part, provide a basis to justify contractor recommendations of support resources. Such decisions must be made as an integral part of the system design (13:2).

As the system design becomes firm in the design and development process, the opportunities to influence component design are reduced and the concentration shifts to using RLA in finalizing maintenance planning and identifying the system's support resource requirements. The maintenance repair level impacts the entire logistics requirements for each new part within the system. For example, the types of support equipment (SE), the training requirements, the technical manuals needed, and supply support levels (i.e., quantities of spares) are all affected by the repair level decision (7:2). Planning for cost effective support with full consideration of the operational alternatives of the weapon system warrants significant attention in making repair level decisions.

Included in this cost effective support planning is the decision to buy test program sets for the failure detection and repair of avionic line replaceable units and shop replaceable units. A test program set (TPS) consists of a test program on a disk, data to execute the test program, and an interface test adapter (ITA) that provides a compatible electrical or mechanical interface between the line replaceable unit (LRU) or shop replaceable unit (SRU) being tested and the automatic test equipment (ATE). Test programs are developed specifically for the LRU or SRU and are used for failure detection and repair by controlling the testing operations and procedures of the ATE (14:11). The failed components in a LRU or SRU are detected using a TPS in conjunction with standard I-level or D-level automatic test equipment and technical orders. These components are then replaced and the LRU or SRU repaired.

Although RLA is conducted throughout the life cycle of a weapon system, many repair level decisions are made early in the design and development phase using development data that may not be valid after production of the system. That is, the decision to repair an SRU might be made many years before the contract to buy the TPS used to repair it is negotiated. In the years between the repair level decision and the TPS acquisition contract settlement, the design of the SRU may change, thereby influencing its repairable characteristics and perhaps increasing the price for the TPS. It is in this area that this research should be of interest to acquisition managers because it provides a means for determining support costs throughout the component development process.

Network Repair Level Analysis (NRLA). Mathematical models have been developed for use in comparing the relative economics of various repair options. Specific RLA models or techniques vary between Department of Defense (DOD) services, but they all use the same basic approach of computing and summarizing support costs. The models compare each repair and discard alternative and identify the least cost solution (23).

NRLA is a computer model developed by the Air Force Acquisition Logistics Center (AFALC) at Wright-Patterson Air Force Base (WPAFB), Ohio. NRLA has been chosen by the Air

Force as the preferred method for making repair level recommendations for avionics and engine components. This model is used by the Air Force as well as the Army, Navy, and over 100 contractors on contract with the Air Force (23).

In order to make recommendations, NRLA uses life cycle costs (LCC) associated with each repair level alternative. NRLA is not, however, a comprehensive LCC model. It does not attempt to include all LCC elements. It only includes those costs that directly affect repair level decisions (3:7; 15:2). The NRLA repair level recommendations are used in conjunction with non-economic factors such as safety, vulnerability, deployment and mobility requirements which can affect the repair level decision.

NRLA is the Air Force preferred means of performing RLA because it formulates the repair level relationships between LRUs, SRUs, and their related support equipment. It solves the repair level problem using failure modes to recognize that an LRU may fail in any of several different ways. The model treats the individual failure modes as part of the overall problem. By structuring and solving the problem as a network, NRLA achieves a system-wide solution to the repair level problem (3:1). As well, NRLA possesses the capability to perform sensitivity analyses to investigate certain "what-if" situations regarding repair level decisions.

NRLA is written in the FORTRAN programming language and was developed on the Air Force Logistics Command (AFLC)

Honeywell 635 computer (5:1). It is available for use on the VAX™ 11/780 as well as on MS-DOS® based micro-computers.

As it exists today, NRLA is a cumbersome, work-intensive model to use. The lack of a good user interface makes entering and editing data a difficult process. Output analysis can be troublesome as well. Although the output products of the NRLA model are useful to planners and managers, the model's design does not provide for easy model manipulation.

Currently there exists no easy way for maintenance planners and acquisition managers to make use of the NRLA model. Because the model is written in FORTRAN, the user must enter data into unformatted input files ensuring that the data is correctly organized for the NRLA model to use. This procedure is quite tedious, very time consuming, and very susceptible to input and editing error. Output can be obtained in a number of printout formats; however, in order to analyze cost figures in the acquisition of TPSs, data must be extracted from numerous printouts. This data must then be entered into another database for determining breakeven costs.

Decision Support Systems. A decision support system (DSS) can aid managers and planners in making critical decisions concerning highly unstructured problems. A DSS can process large volumes of information and can be capable of analyzing complicated issues and relationships (10:12).

While many definitions of a DSS exist in the literature, they all emphasize the following characteristics: a computer based system that helps decision makers effectively confront ill-structured problems through direct interaction with data and analysis models (18; 25:1). To this end, a decision support system is a computer system aimed at quickly assisting managers in making effective decisions in those areas where both management judgment and computer analysis are required (6:129). Carrying this definition one step further, a specific DSS is the hardware and software that allows a specific decision maker or group of decision makers to deal with a specific set of related problems (25:12). A DSS made up of a dialog base, a database, and a models base can provide great insight into a specific problem.

The dialog base of a DSS acts as an effective interface between the user and the other portions of the DSS. This user interface component is responsible for providing all interaction and communication between the computer and the manager (10:81). The database component contains the data needed to describe the problem under investigation. A database management system (DBMS) is required to store, retrieve, and format information to and from the database. This DBMS can be an independent component or can be included as an integral part of the DSS (10:77). The models base is the most complex portion of a DSS. This component provides a mathematical representation of the complex structure and relationship between various elements of the problem under analysis (10:77).

Specific Problem

Easier methods of entering and editing data and obtaining breakeven costs from the NRLA model are desired by maintenance planners and acquisition managers (23; 8). A decision support system needs to be developed in order to enhance the usefulness of the NRLA model. The development of a DSS will ease the data entry and editing process by providing a user interface and database management system that remove the burden of manipulating unformatted input data files. The integration of this interface with a database management system will enhance the total usefulness of NRLA. Incorporation of a model that calculates TPS breakeven costs will provide an additional capability. A specific DSS is needed which enables the user to manipulate the NRLA model in deriving TPS breakeven costs (8).

Importance of Research

To be effective, RLA must be an iterative process that is geared to a schedule and proceeds along with the total systems engineering effort. Timely government review of contractor provided RLA information can be critical. When RLA is unnecessarily delayed due to program stringency, functional areas such as support equipment (SE) development are forced to begin work without benefit of an iterated RLA

(1:1, 2:1). This can cause a great delay in the support resources development process.

Through the development and integration of this DSS using NRLA as the models base, an easy to use and efficient system will be available for the maintenance planner and acquisition manager to obtain verification that the contractor's computer model for RLA during design and development is valid before RLA is fully accomplished and other support activities complete design. This DSS will also facilitate RLA reiteration through the full scale development (FSD) and production phases of the weapons system. The repeated accomplishment of RLA must occur when elements of a system such as mean time between failure (MTBF), mean time between demand (MTBD), unit price, SE, or SE unit price change to affect the repair level decision.

This research is most useful to planners and managers because of the iterated accomplishment of the NRLA model that is required. As stated earlier, certain repair level decisions are made long before the acquisition process for support resources begins. Therefore, acquisition managers need an easy way to obtain cost guidelines in order to buy the required resources (i.e., TPSs). Through an innovative use of the NRLA model, the TPS Max Cost model provides the capability to obtain these guidelines easily.

Figure 1 shows graphically how the TPS Max Cost model works. Shown is a typical TPS cost curve. For illustration, assume that the curve displayed is for an SRU TPS that costs $200,000. Units for the X-axis are SRU unit costs which increase from the origin. Units for the Y-axis are SRU MTBDs which decrease from the origin. The curve represents the various combinations of component reliability and unit cost associated with a TPS cost of $200,000.

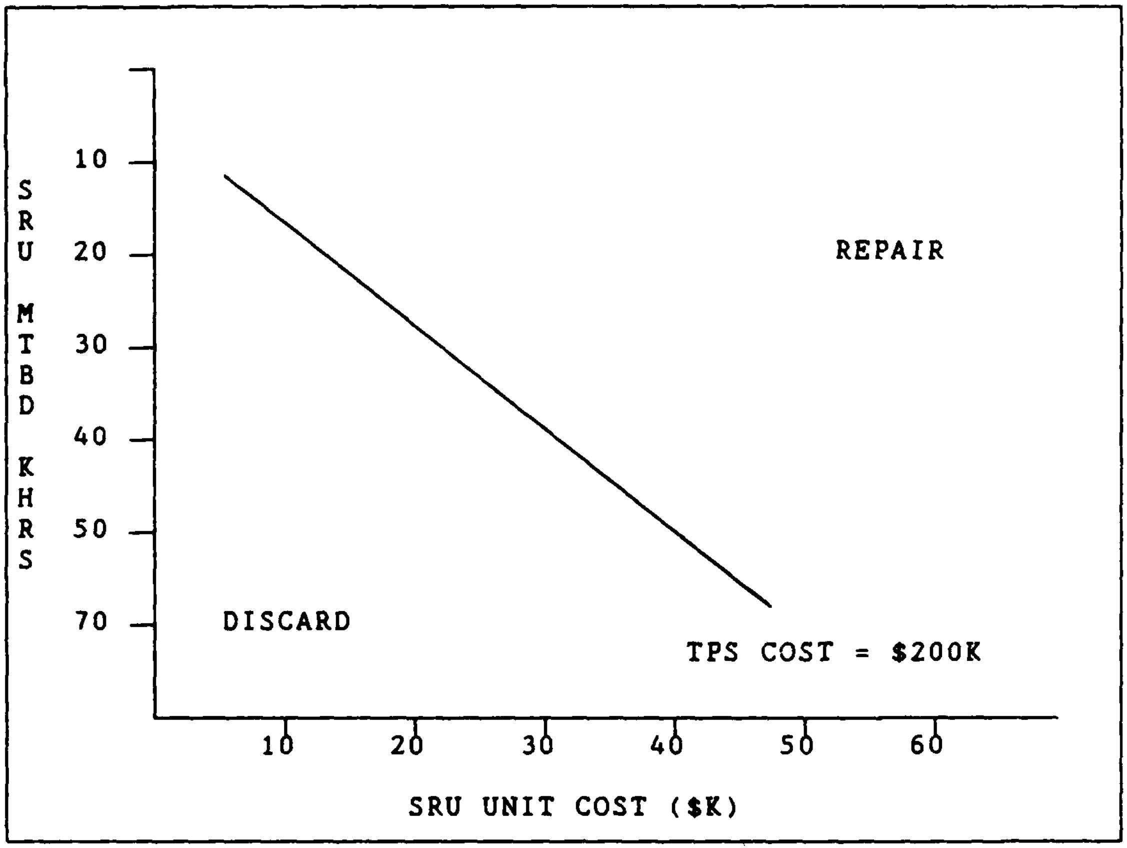


Fig. 1. SRU TPS Cost Curve

The intersection of the component's reliability and unit cost shows the breakeven point for acquiring the TPS. If this intersection is above and to the right of the TPS cost curve (i.e., high unit cost and low reliability) then it is cost effective to buy the TPS because the proposed TPS cost is below the breakeven point. If the intersection is below and to the left of the curve (i.e., low unit cost and high reliability), it is not cost effective to buy the TPS (17).

This model can be quite useful when negotiating multi-million dollar contracts. Acquisition managers can use this portion of the DSS as a negotiating tool. It can be used to motivate contractors to increase the component's reliability or decrease its TPS cost in order to get the contract.

Research Objectives

In order to ensure that an efficient, easy to use and meaningful system is put to use, the needs of the system and system users were identified. Once a full description of the users' needs was obtained, this research effort worked toward producing a decision support system that fulfills and supports those needs.

Research Questions

In order to determine the feasibility of developing the NRLA model into a specific DSS and to produce a system that is both useful and meaningful, many investigative questions were answered. The following questions were considered in this process to provide insight for system development:

1) How can the development of a DSS using NRLA as the models base enhance the repair level analysis process?

2) What similar software systems are being used elsewhere in the Air Force to meet similar

requirements? If software systems/modules are available, can they be adapted for use here?

3) If there are no adaptable systems available, could a system be developed that will make use of the existing model?

4) If writing a procedure for the modules,

   a) What are the specific NRLA compatibility requirements for the system to work successfully?

   b) Which commercial software packages will best satisfy these requirements?

5) What criteria and concerns must be considered to ensure successful acceptance and implementation by the users?

Scope and Limitations

This research effort was limited to the development of an initial specific decision support system that improves the usefulness of the NRLA model. This development process focused on providing a means to ease the data manipulation process required by the NRLA model as well as expanding its applicability for use in the acquisition arena.

A user interface enables the user to create and edit data files, run the NRLA and TPS cost models, and provides access to all NRLA functions including the existing output mechanisms. The addition of the TPS acquisition cost model provides guidelines regarding the maximum cost the government should pay in procuring TPSs. These software modules do not alter any internal functions of the NRLA model to affect its existing intended use.

Portions of the text contained in this chapter were taken from a term paper on the same subject written by Girz and Romrell (21:et passim).

II. Methodology

Objectives

The desired output of this research was an initial specific DSS made up of a set of micro-computer based programs. This system could easily be loaded and run on an MS-DOS(R), IBM(R) compatible system with a hard drive. Program documentation and a users' guide are included to provide the user the necessary insight into system construction and use.

Developing the Decision Support System.

A four-step procedure was developed in order to meet the research objectives and answer the research questions. The first step included an initial assessment of the NRLA model for its suitability for development into a specific DSS. This preliminary study included a literature review of DSS characteristics as well as a review of NRLA documents to determine research applicability. The next step included developing the DSS environment by assessing the existing environment, identifying and interviewing the DSS group, and identifying the necessary DSS tools. The third step involved the development, test, and validation of the initial specific DSS. The final step included the identification of potential subsequent specific DSS developments (24:67-69). This is covered in Chapter V.

Preliminary Study. The literature review of this preliminary study was limited to determining the requirements for a system to be considered a decision support system. It was also concerned with determining parameter requirements in order to develop a system that will be compatible with the NRLA model.

In order to determine if the NRLA model can be used in the development of an effective and useful DSS, an extensive review of the NRLA model documentation was accomplished. This documentation was available from the Directorate of Plans/Studies and Analysis (AFALC/LSX), and included the *Network Repair Level Analysis Model User's Guide*, the *Network Repair Level Analysis Instant Expert Guide*, and the *Network Repair Level Analysis Model Programmer's Guide*. These documents describe the model's program structure, logic, input and output operations, and the organization of data used in the program so that modification or corrections can be made (3:1; 4:1; 5:1). This review gave insight into the feasibility of this research as well as providing the necessary information regarding parameter requirements for the user interface.

Developing the DSS Environment. This step involved assessing the existing environment, putting together the decision support group and gathering information, and developing criteria for determining the correct DSS tools to use in developing the initial DSS.

The Existing DSS Environment. Further research into the NRLA model was done to identify shortfalls and solutions in order that DSS development could begin. This review assessed the existing environment to determine shortfalls of the dialog base, the database, the models base, and the output generator.

Decision Support System Group. In assessing the appropriate DSS development group, maintenance planners, acquisition managers, and personnel from the NRLA staff office (AFALC/LSX) were considered. This group of people had expressed an interest in this development process, were available at the AFALC at Wright-Patterson Air Force Base, and were consulted throughout the development process.

Maintenance planners make use of the NRLA model throughout the weapons system development process on an iterative basis. Their expertise was essential in determining data input parameter requirements. Acquisition managers suggested that a means of gathering information for negotiations for acquiring test program sets be made available and were consulted to ensure their requirements were being fulfilled.

The NRLA staff office (AFALC/LSX) is the office of primary responsibility (OPR) for the NRLA model and must concur with any changes or enhancements to the model (23; 16:3). Coordination with the NRLA staff office provided insight into the structure of the NRLA model. Their help ensured the proper integration between the user interface, the database, database management facilities, and the analysis models base.

Close coordination with this DSS group was kept in order to obtain user feedback throughout system development. This contact was initiated through interviews and was continued through informal technical interchange meetings.

Interviewing. Personal interviewing was chosen as the primary method for gathering information regarding the identification of the users' needs and requirements. This type of information gathering technique has some real advantages as well as disadvantages. However, it was chosen because of the depth and detail of information that can be obtained from the respondents.

Due to bias that is introduced by the interviewer during questioning sessions, interviewing can place limits on the information gathering process. The respondent's subject knowledge as well as his willingness to respond can have an effect on the information gathering (9; 19:160). Cost can be a factor as well.

Although these are real concerns, their effects on the quality of the information gathered were minimal. Interviewer bias did not have a big effect because the interview premise was to gather information about how the development of this DSS could best benefit the user and not to sway the respondents' feelings on a particular subject. Cost was not a factor because all of the interviews were conducted at Wright-Patterson AFB.

According to both Davis and Emory, the information received during interviewing is of a better quality than information gathered by other techniques because the interviewer has more control over the information gathering process (9; 19:160). Therefore, unstructured personal interviews and technical interchange meetings (TIM), in particular, were used to obtain the necessary information.

The interview process continued throughout the system development process to ensure that the users' needs were fulfilled. These interviews and TIMs were held with personnel from various offices on the AFALC staff, as well as with logistics personnel from Aeronautical Systems Division's (ASD) System Program Offices (SPO) at Wright-Patterson AFB.

Decision Support System Tools. In order to develop this DSS, criteria for determining the correct tools to use was required. These tools are the hardware and software elements which facilitate the development of a specific DSS (25:13). Research into the appropriate hardware, software, and database management systems (DBMS) was conducted to determine the most applicable set of tools available. The application of these tools provides a system that fulfills user requirements and is efficient and easy to use.

Hardware. The type of computer facility needed to process the data depends on the required storage capacity and how the information is entered in the computer (10:77). An investigation into the typical data and software storage requirements was conducted. The only other requirement for the hardware was that it must be presently in the Air Force inventory and available for use at the AFALC and ASD. No new hardware was to be acquired in order to make use of this DSS.

Software. Since the software ties together the various parts of a computer system to meet the needs of each application (10:143), it must be both flexible and easy to use. It must provide access to the user interface, the operations of the database management facility, and the models base. Research to gather information regarding the applicability of commercial as well as existing Air Force software systems was performed. This included reviewing software users' guides and handbooks as well as reviewing related theses. Through interviews with personnel on the AFALC staff, existing Air Force-used software was investigated for use in this research.

Database Management System. The database management system (DBMS) must be capable of interfacing with the other DSS components. It must enable the user to easily modify data inputs, sort the data, and provide a data verification capability to ensure that the proper data type is in the proper fcrmat for model use.

Developing the Initial Specific DSS. Development of the initial specific DSS was accomplished by breaking down the DSS structure into four subsystems and interactively tying those subsystems together. These four subsystems are the user interface, the database, the DBMS, and the analysis models. Once the requirements of the initial specific DSS were determined, a prototype was developed and installed at the NRLA staff office (AFALC/LSX). The requirements determination for the subsystems as well as the prototype development of the DSS are discussed in Chapter III.

Test and Validation. Test and validation of the final product was accomplished using data known to give specific output from the NRLA model. This consisted of loading data through the user interface and database management system and running the NRLA model. The generated output was compared with output resulting from running the model using data not entered through the applications programs developed here. The data used for test and validation was available from AFALC/LSX (23). This also is discussed further in Chapter III.

Portions of the text contained in this chapter were taken from a term paper written on the same subject by Girz and Romrell (21:et passim).

III. Determinations and Discussion

Preliminary Study

As defined in Chapter I, a decision support system helps decision makers effectively confront ill-structured problems through direct interaction with data and analysis models. Careful selection of item repair policy can be considered an ill-structured problem that requires the interaction with data and analysis models.

Because NRLA is an analysis model that confronts the ill-structured, repair level problem, it was determined that it could be used in the development of an effective decision support system. It was decided that with the development of a user interface and a DBMS, the NRLA model could be integrated into a useful specific DSS. In order to do so, however, the needs of the users had to be determined.

Maintenance planners and acquisition managers at the AFALC felt that the use of the NRLA model could be expanded (8; 23). Improvements to the user interface and the models base would increase the usefulness of NRLA by improving user/model interaction and providing greater insight to unstructured repair level and support acquisition problems (8).

Two basic problems hinder the effectiveness of this system. First, the lack of a good user interface complicates the entering and editing of data; and secondly, the present model does not analyze breakeven costs. The output products of the NRLA model are useful to planners and managers; however, the model design does not provide for further data manipulation in calculating breakeven costs (8).

A dialog feature to enhance human interface would greatly improve the use of the model for iterative computations and sensitivity analyses. This would increase NRLA's usefulness to acquisition managers. The development of a user interface module would simplify the data entry and editing process. This dialog feature would remove the burden of direct data file manipulation. Through integration of the user interface and the DBMS, the data would be placed in the correct field within the data file for NRLA to use. The integration of the TPS Max Cost model would enhance the analysis capabilities by providing the capability to calculate break-even costs for use in the acquisition of TPSs.

Developing the DSS Environment

The Existing DSS Environment. The NRLA model provides support in making repair level decisions. As it exists, the model possesses characteristics of a DSS. It helps decision-makers effectively confront the ill-structured problem of achieving a cost-effective repair policy for components of a weapons system by considering the relevant cost factors. It does this through direct interaction with database files drawing from over 110 data points for use in the model.

Its only pitfalls are the lack of a good user interface and DBMS and the lack of ability to analyze TPS acquisition cost estimates. Development in these areas hoıds the most potential for improving the NRLA model and expanding its usefulness.

Database. As it exists, NRLA does not make use of a database management system. In order to run the NRLA model, the user must enter data into two input data files. The Program Specific Data File contains all the relevant cost information for the weapon system under analysis. The data in this file are unlikely to change from run to run. The Run Specific Data File contains information that runs the model according to the sensitivity analysis alternatives entered in this file by the user. It also contains information regarding the type of output desired.

The data within these two input data files are categorized by and separated into 11 data record types as listed in Table I. The information within the data record types are stored in data records. Data record types 1, 2, 3, 9, and 10 will always contain only one data record each. Data record types 4, 5, 6, 7, 8, and 11 may contain multiple data records within each. This distinction between data record types and data records is important and is discussed further in Chapter IV.

Because of the possibility of multiple data records this model can draw from well over 110 data points. On highly complex equipment the number of data points can get quite large depending on the number of LRUs, SRUs, and the number and types of SE required (5:17-34).

Table I. Data Record Types

Program Specific Data

Record Type No.	Record Type Name
1	Weapon System Data
2	Maintenance System Data
3	Supply System Data
4	Support Equipment Data
5	LRU Description Data
6	LRU Failure Mode Data
7	SRU Description Data
8	LRU/SRU/SE Cross Reference Data

Run Specific Data

Record Type No.	Record Type Name
9	Output Options Data
10	Wholesale Factors Data
11	Individual Change Factors

Analysis Models Base. The NRLA model uses life cycle costs with LRU/SRU/SE interdependency relationships to construct a network representation of the repair level decision problem. Through the use of this network, the model explicitly considers indenture level relationships between LRUs and SRUs. It solves the problem by failure mode, recognizing that a LRU may fail in any of several different ways, and treats the individual failure modes as part of the overall problem (3:21). Through this network formulation, a structural tie is developed in a way that permits the network algorithm to select only the SE resources which are economically justified and which minimize overall costs.

To illustrate this network formulation, Table II shows LRU/SRU/SE relationships for three different LRUs. The first LRU, identifier 63A00, has no SRUs associated with it. The second, LRU 63D00, has two failure modes. SRU 63DA0 is associated with one of those failure modes. LRU 63J00 has three failure modes--all with associated SRUs. The Xs in the columns to the right show which type of support equipment is required to repair the LRUs and SRUs depending on the failure mode.

Table II. LRU/SRU/SE Relationships (3:32)

LRU IDENT	FM #	SRU IDENT (if any)	Depot SE				Intr SE			
			1001	2002	2003	3003	5001	6002	6003	7005
63A00	1		X				X			
63D00	1				X				X	
	1	63DA0			X				X	
	2				X	X		X	X	X
63J00	1		X				X			
	1	63JA0		X	X			X	X	
	2		X				X			
	2	63JB0		X	X			X	X	
	3		X				X			
	3	63JC0		X	X			X	X	

For example, for LRU 63D00, the first failure mode involves SRU 63DA0 and the second failure mode has no SRU associated with it. To remove and replace SRU 63DA0, SE 2003 is needed at the depot, and SE 6003 is needed at the intermediate level. To repair the SRU, the same SE is required. If the second failure mode occurs, both SE 2003 and SE 3003 are required at the depot to repair the LRU. SE 6002, SE 6003, and SE 7005 are required to repair the LRU at the intermediate level.

It is of interest that the LRU/SRU/SE relationship table is translated into a network by the computer, enabling the system to be solved as a network. There must be an entry in the SE Cross Reference Data Record Type for each horizontal entry in this table (3:23).

Output Generator. There are 11 types of output reports that can be generated by the NRLA model. These are listed in Table III.

Table III. NRLA Output Reports (5:35-42)

General Information
SE Input Values
SE to LRU/SRU Relationships
Computed SE Costs
SE Requirements
LRU and SRU Repair Level Decisions
LRU Repair Location Summary
Repair Level Decision Details
Summary Sensitivity Analysis Results
Summary Statistics
Detailed Sensitivity Analysis

The output contains a printout of the input data, intermediate results, supplementary information, the optimal solution, and sensitivity analysis results. Upon analysis completion the model solution is displayed on the screen. The Output Options Data Record Type allows the user to choose the desired output.

Sensitivity Analysis. A major attraction of the NRLA model is its capacity to perform a sensitivity analysis. This analysis is performed to investigate effects due to incorrectly estimated data, to process changes, or to answer "what-if" questions. The analysis may then be used as rationale for improving estimates, selecting projects for further developmental work, or doing nothing, as the particular situation warrants.

The analysis is performed by selecting a relevant range. For example, the repair level decision could be changed when the mean time between failure (MTBF) of a component varies between 50 percent and 200 percent of the input baseline MTBF. Four analysis procedures provide the user with information about how the repair level decision changes over this range.

Decision Support System Group. As identified in Chapter II, the DSS group should be made up of acquisition managers, logistics planners and personnel from the NRLA staff office (AFALC/LSX). These were the types of people necessary for coordination in this development effort. This group must be familiar with the NRLA model. Although this DSS will greatly improve and ease the use of the NRLA model, the users must be able to understand what the required data means in order to effectively use this DSS. The desk reference and program documentation provided herein by no means replace the *Network Repair Level Analysis User's Guide* available from AFALC/LSX. Inexperienced NRLA model users should consult the *Network Repair Level Analysis User's Guide* in conjunction with the desk reference provided by this research effort when working with this DSS.

Decision Support System Tools. Following the development of criteria for determining the DSS tools to use in the development of the DSS, the following decisions were made.

Hardware. A micro-computer is the ideal computer environment in which to operate this DSS. A micro-computer with a 20 megabyte hard drive such as a Zenith Z248$^{(R)}$ will provide enough memory storage to access the user interface, DBMS, database files, and analysis models without the burden of working around a floppy drive system. dBase III Plus$^{(R)}$ uses 600,000 bytes of the hard drive with the software applications programs, including analysis models, using 500,000 bytes. A color monitor is also desirable but not required for this DSS.

Software. The software package chosen for the development and integration of this specific DSS is dBase III Plus$^{(R)}$. Other software packages considered for use were dBase III^{TM} and Turbo Pascal$^{(R)}$. However, because of its database management capabilities that can be integrated with applications programs, dBase III Plus‘*’ was determined to be the best choice. dBase III Plus‘*’ is also available within the Air Force inventory and used widely. Application programs were written in dBase III Plus‘*’ and make up the menu-driven, user interface. These programs were integrated with a database management facility that provides access to the analysis models base without the user having to exit the dBase environment. Through the use of dBase III Plus‘*’, the user interface, the database, the DBMS, and the analysis models components were linked together and a totally integrated DSS was achieved.

Data Base Management System. As described earlier, NRLA presently makes use of two data input files in drawing data for analysis. This does not permit easy data management. Through application programs written in dBase III Plus‘*’ an integrated DBMS is provided to perform the necessary data management and ease the data manipulation requirements.

Developing the Initial Specific DSS

Structure. Development of the initial specific DSS was accomplished by breaking down the DSS structure into the following four subsystems: user interface, DBMS, database, and analysis models. These subsystems are totally integrated through the use of dBase III Plus‘*’. In certain cases it is hard to distinguish between the user interface and the DBMS because some programs perform functions of both subsystems. This is due to the integrated nature of dBase III Plus‘*’.

The DSS is made up of four modules which can be accessed from the Main Menu as shown in Figure 2. Through these modules, the user can display data, edit data, create new data records or run the analysis models.

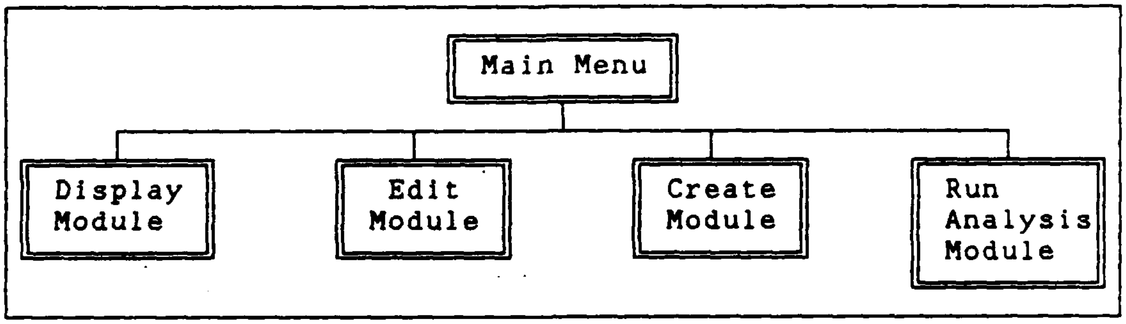


Fig. 2. DSS Module Organization

The applications programs written that make up the user interface and the DBMS provide the user access to each of these modules. The data is stored in 13 database files from which the DBMS programs draws information and writes it to program specific and run specific files as named by the user for NRLA model use. The dBase III Plus‘*’ environment allows program control to pass from the applications programs developed here to the NRLA and TPS Max Cost analysis models. The NRLA model will draw data from the program specific and run specific files to which the DBMS applications programs had written data. The user must input data for the TPS Max Cost model. Upon analysis completion, program control is automatically passed back to the applications programs that make up the user interface and the DBMS for further data manipulation. Chapter IV describes in detail the operations necessary to effectively use this DSS.

User Interface. The user interface consists of 48 programs coded in dBase III Plus‘®’. Certain programs perform both user interface functions and DBMS functions as well. A series of menu driven prompts query the user and move him through the various system modules to perform the required data manipulation and analysis functions. System structure and use is described in detail in Chapter IV, Software Documentation.

Database Management System. The database management system makes interactive use of the NRLA model much easier. This permits iterative input data manipulation for sensitivity analysis and data editing due to weapons system parameter changes. As previously stated, the DBMS ensures that the data entered by the user is in the correct format for the analysis models to use. Chapter 2 of the *Network Repair Level Analysis User's Guide* describes in detail the specified format for the data fields that are required in the program specific and run specific input data files. The applications programs that make up the DBMS take into account these requirements including the floating decimal point format that the NRLA model follows when reading input data contained in these data files. Application programs written in dBase III Plus‘R’ provide access to all DSS components, permitting data storage, retrieval, and manipulation such as sorting, editing, and deleting.

Database. dBase III Plus‘R’ itself acts as the database facility for this DSS. Data can be entered and stored in 13 database files. There is a database file for each of the 11 data record types listed in Table I. The Support Equipment Data Record Type has two database files associated with it because the data records contained in this data record type must be sorted in a particular manner for use by NRLA. The applications programs that make up the DBMS retrieve data records from the primary database file (SE_DATA.DBF) and sorts them by SE Resource Number. The sorted data records are then written to the secondary database file (SESORT.DBF). NRLA then uses the data contained in SESORT.DBF. The thirteenth database file contains data pertaining to the date of the NRLA run in month, day, and year.

The structures of the 13 database files are shown in Appendix A. Database files for different subsystems of weapons systems can be stored and retrieved for future manipulation and analysis. This is further explained in Chapter IV.

Analysis Models. The analysis models base consists of two models. The NRLA model performs the repair level analysis as explained above. The TPS Max Cost model provides information regarding break-even costs for the acquisition of

TPSs. This model relies on reliability and unit cost data provided by the NRLA model output for a specific LRU or SRU. It manipulates this data in order to produce cost information representing the most the government should pay for a TPS at certain levels of component reliability and cost. Through a comparison of various component unit cost and mean time between demand ratios the model computes the maximum that the Air Force should pay for acquiring a TPS. This model was originally developed and available for use in determining SRU TPS costs for the B1-B program (17). The TPS Max Cost model used in this initial specific DSS should only be used to determine costs for the B-1B program.

As program control is passed to these models, the analysis is performed using data written to user specified data files in the case of NRLA or using data provided by the user in the case of the TPS Max Cost model.

Output Generator. Output from the NRLA model, as described above, can be produced in 11 different data formats. As such, there is no output generator per se. The user can specify the type of output he desires to be generated through the Output Options Data Record Type. This portion of the DSS was not enhanced and makes use of the existing facility provided by the NRLA model.

The TPS Max Cost model provides information regarding the maximum the government should pay for acquiring a TPS. It displays the maximum cost the government should pay in procuring a TPS for the LRU or SRU under analysis. The output of this model is displayed directly on the screen and can easily be sent to the printer for a hard copy.

Initial Specific DSS. Following the determination of requirements for the development of an initial specific DSS, a prototype was developed and installed at the NRLA staff office (AFALC/LSX). AFALC/LSX has a Zenith Z248$^{(R)}$ PC with a 20 megabyte hard drive and color monitor that fully utilizes the system's integrated software.

The initial DSS consisted of 68 dBase III Plus$^{(R)}$ applications programs that make up the user interface integrated with the DBMS and analysis models. A listing of the programs coding is provided in Appendix D.

Test and Validation. Initial test and validation took place using data provided by AFALC/LSX. This data was used to create a new data file and run the NRLA model. The data was created, edited, and stored using the applications programs that make up this DSS. The NRLA model was run using the DSS and the generated output was compared to the output generated by the NRLA model using identical data that was not input using this DSS.

This initial effort uncovered some problems with the applications programs that write the data to the program specific data file for NRLA to use. These initial problems required further programming to ensure that the data was being written in the proper format. Following this, a second test effort resulted in a successful validation of the performance of these applications programs. The NRLA output generated through the DSS was identical to the output generated by running the NRLA model without the use of the DSS.

This initial validation process was performed twice, using two different sets of data having known output figures. The anomalies uncovered by these two tests have been corrected and integrated in the applications programs.

Portions of the text contained in this chapter were taken from a term paper written on the same subject by Girz and Romrell (21:et passim).

IV. Software Documentation

This chapter describes the operations necessary to effectively use this DSS. A step-by-step process is provided in order to create, edit, and display data as well as run the analysis models and obtain results. The actual program coding is shown in Appendix D. This coding used many applications and common techniques of basic dBase III Plus$^{(R)}$ coding (20; 22; 26).

In using this system it is assumed that dBase III Plus$^{(R)}$ is available to the user. The user need only know a limited number of commands in order to use this system. A user's desk reference is provided in Appendix G which gives the necessary insight into system operation.

Here and subsequently in this chapter, the term "the system" refers to the applications programs compiled to make up the DSS. Any other use of "system" will be specified as such (i.e., weapons system, maintenance system, disk operating system, etc.).

Installing the System.

Before starting dBase III Plus$^{(R)}$, the disk or directory used to start the computer should contain a CONFIG.SYS file. This file should contain the commands:

```
FILES = 20
BUFFERS = 15.
```

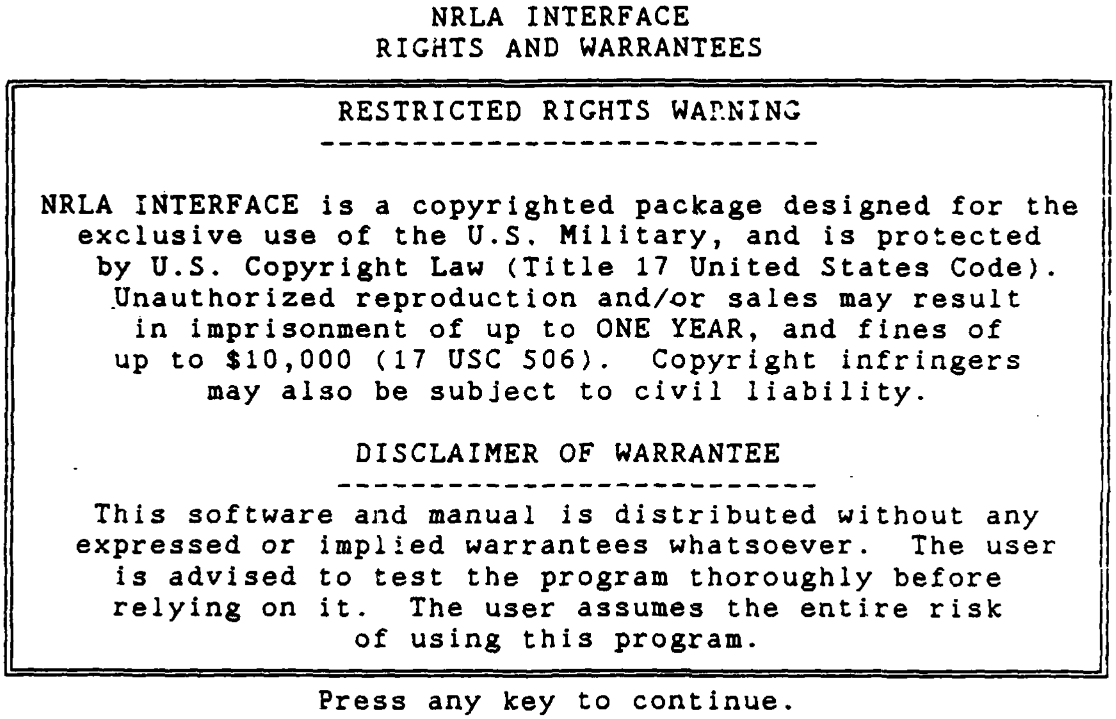
The CONFIG.SYS file is a text file that can be created with any word processor that creates ASCII text files. If this file is not present, dBase III Plus$^{(R)}$ will operate, but may encounter a "too many files are open" error message. The only cure is to place the CONFIG.SYS file on the disk used to start the computer (22:3).

To effectively use this system, dBase III Plus$^{(R)}$ must be loaded in a separate subdirectory on the PC hard drive. The 68 applications programs, the 13 database files, the NRLA model, and the TPS Max Cost model should also be loaded into this directory.

Starting the System

In order to start NRLA INTERFACE, which is the portion of the DSS that acts as the user interface of the DSS, the user must enter the subdirectory containing the above mentioned programs and enter the dBase III Plus$^{(R)}$ system. At the dot prompt within dBase III Plus$^{(R)}$ the user need only enter the command "DO NRLAFACE" to start the NRLA INTERFACE program and the DSS.

The first screen the system displays is the Rights and Warrantees screen as shown in Figure 3. This screen outlines the copyrights and warrantees associated with using this software package. This screen is displayed until a key is pressed by the user. The system then brings up the Main Menu Screen which enables the user to manipulate the system allowing him to enter the display, edit, create, or analysis modules.



NRLA INTERFACE
RIGHTS AND WARRANTEES

RESTRICTED RIGHTS WARNING

NRLA INTERFACE is a copyrighted package designed for the exclusive use of the U.S. Military, and is protected by U.S. Copyright Law (Title 17 United States Code). Unauthorized reproduction and/or sales may result in imprisonment of up to ONE YEAR, and fines of up to $10,000 (17 USC 506). Copyright infringers may also be subject to civil liability.

DISCLAIMER OF WARRANTEE

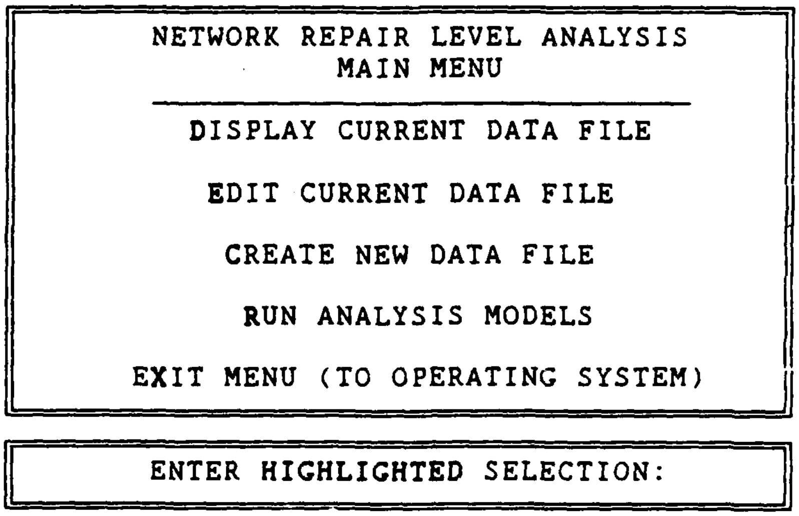
This software and manual is distributed without any expressed or implied warrantees whatsoever. The user is advised to test the program thoroughly before relying on it. The user assumes the entire risk of using this program.

Press any key to continue.

Fig. 3. Rights and Warrantees Screen

Main Menu

The Main Menu Screen shown in Figure 4 is the highest level of operation in this DSS. On a color terminal the Main



NETWORK REPAIR LEVEL ANALYSIS
MAIN MENU

DISPLAY CURRENT DATA FILE

EDIT CURRENT DATA FILE

CREATE NEW DATA FILE

RUN ANALYSIS MODELS

EXIT MENU (TO OPERATING SYSTEM)

ENTER HIGHLIGHTED SELECTION:

Fig. 4. Main Menu Screen

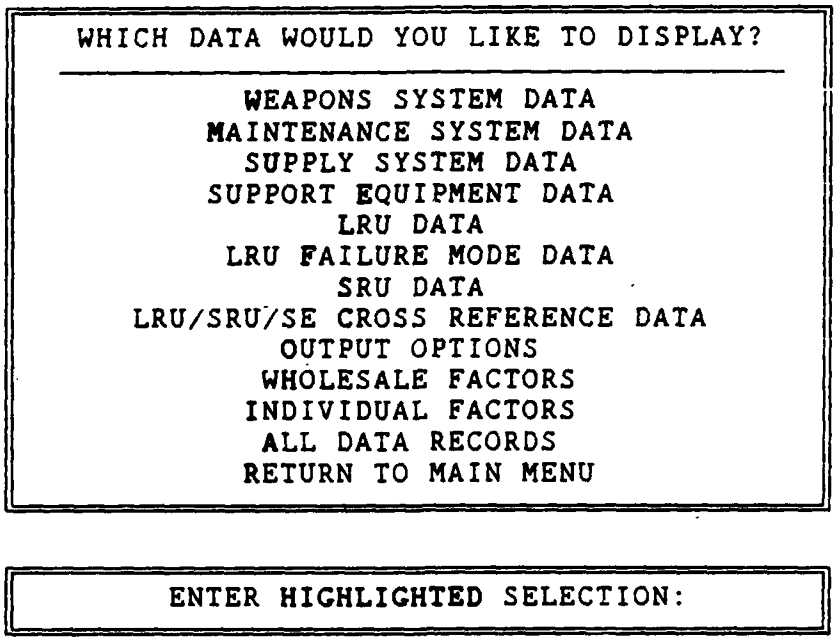
Menu Screen displays the characters of the module choices in blue with the highlighted characters displayed in red. A monochrome terminal displays the characters of the module choices in the screen's color on a black background with the highlighted characters displayed in inverse colors. This pertains to all highlighted characters throughout the system.

In order to enact the operations of any of the four modules the user need only enter the highlighted letter of his selection. The user must ensure that the Caps Lock key is engaged in order for the system to recognize a correct keystroke entry.

At the Main Menu the user has a choice of the following keystroke entries, which here and subsequently are indicated by their inclusion in pointed brackets (<...>): <D>, <E>, <C>, <R>, or <X>. With these entries the user can display, edit, or create data files, run analysis models or exit the DSS to the computer's disk operating system. The system will only accept the above responses at the Main Menu; any other entry will not be recognized. This screen will continue to be displayed until one of these selections is entered.

Displaying Data

A selection of <D> while at the Main Menu selection prompt will bring up the Data Display Screen as shown in Figure 5. The user can enter the highlighted selection associated with any of the 11 data record types to display the data contained within each. The user can only display



WHICH DATA WOULD YOU LIKE TO DISPLAY?

WEAPONS SYSTEM DATA
MAINTENANCE SYSTEM DATA
SUPPLY SYSTEM DATA
SUPPORT EQUIPMENT DATA
LRU DATA
LRU FAILURE MODE DATA
SRU DATA
LRU/SRU/SE CROSS REFERENCE DATA
OUTPUT OPTIONS
WHOLESALE FACTORS
INDIVIDUAL FACTORS
ALL DATA RECORDS
RETURN TO MAIN MENU

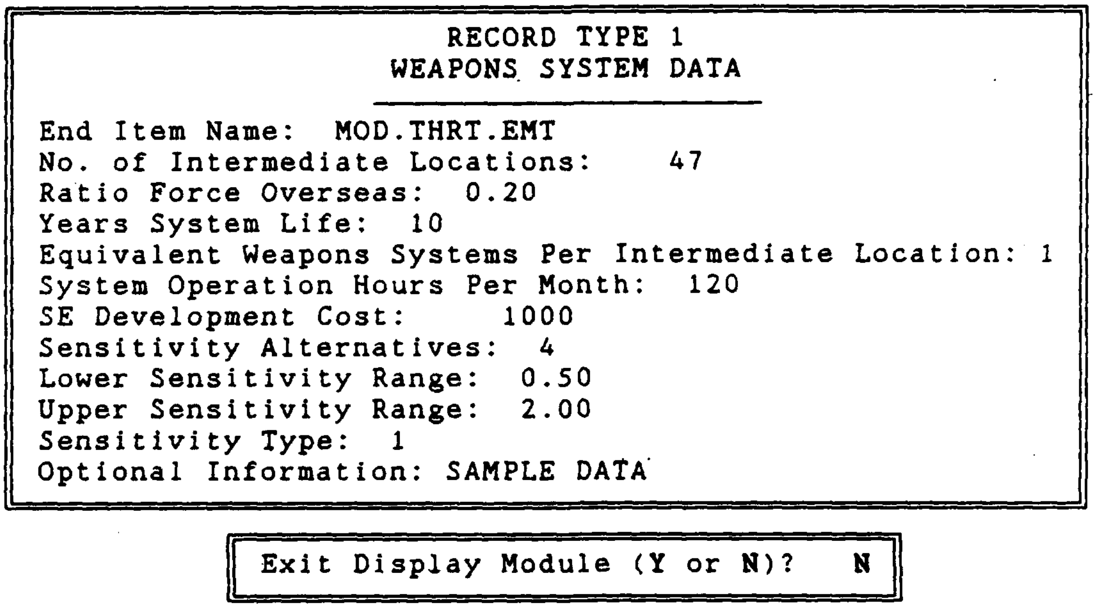
ENTER HIGHLIGHTED SELECTION:

Fig. 5. Data Display Screen

the data through this module; no interaction with the database file is possible. This module provides the user a means to quickly display the data presently stored in the database. It allows the user to verify the correctness of entered data quickly without having to specify individual records to view.

At the Data Display Screen selection prompt the system will only accept the following entries: <W>, <M>, <U>, <E>, <L>, <F>, <S>, <C>, <O>, <H>, <I>, <A>, or <R>. Any other entry will not be recognized and the Data Display Screen will remain on the screen until one of these selections is entered.

Upon entry of a highlighted selection pertaining to a data record type, the system displays the data presently stored in the database file fcr the data record type selected. Figure 6 shows sample data contained in the Weapons System Data Record Type database file. Each of the data record display screens is shown with sample data in Appendix B. A selection of <A> displays all the data stored in each of the data record type database files without having to select individual data record types. A selection of <R> returns the user to the Main Menu.

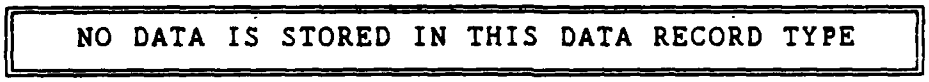


```
                    RECORD TYPE 1
                 WEAPONS SYSTEM DATA

End Item Name:  MOD.THRT.EMT
No. of Intermediate Locations:    47
Ratio Force Overseas:  0.20
Years System Life:  10
Equivalent Weapons Systems Per Intermediate Location: 1
System Operation Hours Per Month:  120
SE Development Cost:     1000
Sensitivity Alternatives:  4
Lower Sensitivity Range:  0.50
Upper Sensitivity Range:  2.00
Sensitivity Type:  1
Optional Information: SAMPLE DATA

            Exit Display Module (Y or N)?   N
```

Fig. 6. Weapons System Data Display Screen

Upon viewing the data display screen the user has the option of remaining in the Display Module or exiting and returning to the Main Menu. The default to the Exit Display Module (Y or N)? prompt is 'N'. If an <N> is entered, the system displays the next record in the file or returns to the Display Screen if at the end of the database file. Upon entry of a <Y> the system returns to the Main Menu.

If no data is stored in the data record type selected, the system will display the Empty Data Record Type Screen shown in Figure 7.

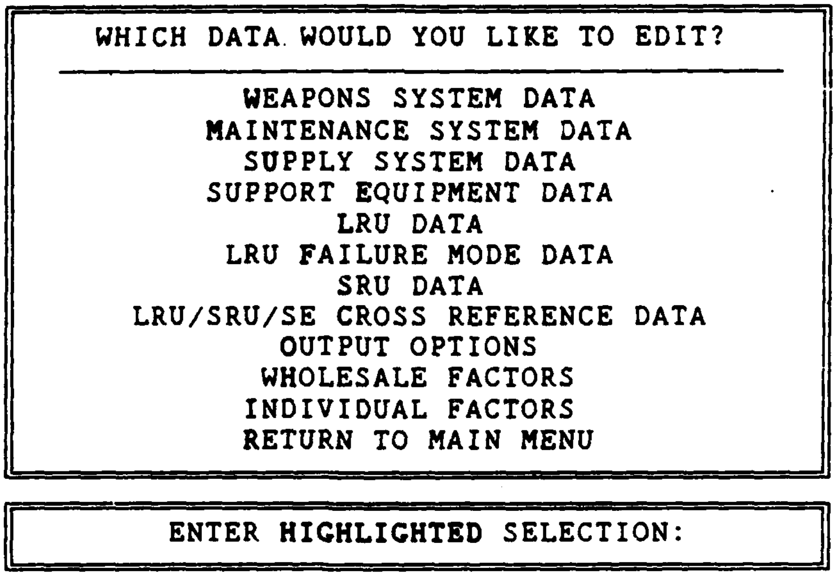


NO DATA IS STORED IN THIS DATA RECORD TYPE

Fig. 7. Empty Data Record Type Screen

Editing Data

A selection of <E> at the Main Menu selection prompt displays the Data Edit Screen shown in Figure 8. At this screen the user can enter the highlighted selection associated with any of the 11 data record types to edit the data contained within each. In order to edit data contained



WHICH DATA WOULD YOU LIKE TO EDIT?

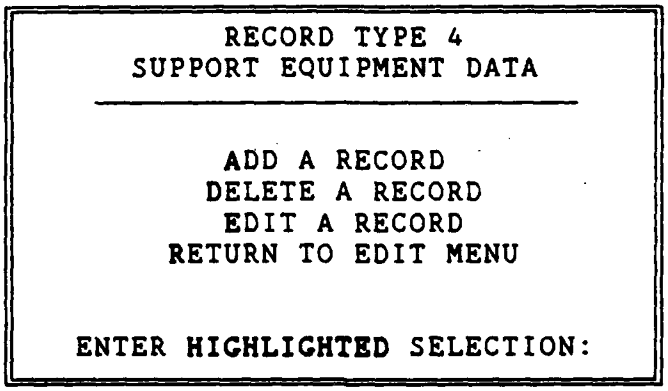
WEAPONS SYSTEM DATA
MAINTENANCE SYSTEM DATA
SUPPLY SYSTEM DATA
SUPPORT EQUIPMENT DATA
LRU DATA
LRU FAILURE MODE DATA
SRU DATA
LRU/SRU/SE CROSS REFERENCE DATA
OUTPUT OPTIONS
WHOLESALE FACTORS
INDIVIDUAL FACTORS
RETURN TO MAIN MENU

ENTER HIGHLIGHTED SELECTION:

Fig. 8. Data Edit Screen

in the weapons system, maintenance system, supply system, output options, or wholesale factors data record types the user need only enter the highlighted character associated with the data record type. These data record types contain only one data record each. Upon the entry of <W>, <M>, <U>, <O>, or <H> the system moves directly to the edit data record screen and allows interaction with the database for editing. Should the user choose not to edit data contained in this record, he may step through the fields on the screen by pressing the <return> key. Pressing <pg dn> will exit the data record edit screen without changing further any data within the database file not previously edited. Any data edited prior to pressing <pg dn> will be saved in the database file as edited upon pressing <pg dn>. Each individual data record edit screen looks similar to the data record display screens for the same data record types shown in Appendix B. They differ in that the user may interact with the data record edit screens and change the data contained in each data record.

Edit Choice Menu. In order to edit data contained in the support equipment, LRU, LRU failure mode, SRU, LRU/SRU/SE cross reference, or individual factors data record types, the user would enter the appropriate highlighted character. Upon selection of <E>, <L>, <F>, <S>, <C>, or <I> the system displays the Edit Choice Menu. This enables the user to add, delete or edit any of the multiple data records within each of these data record types. As an example, the Edit Choice Menu Screen for the Support Equipment Data Record Type is shown in Figure 9. Each of the data record types has a similar Edit Choice Menu in the editing mode. They differ only in their data record type headings.



RECORD TYPE 4
SUPPORT EQUIPMENT DATA

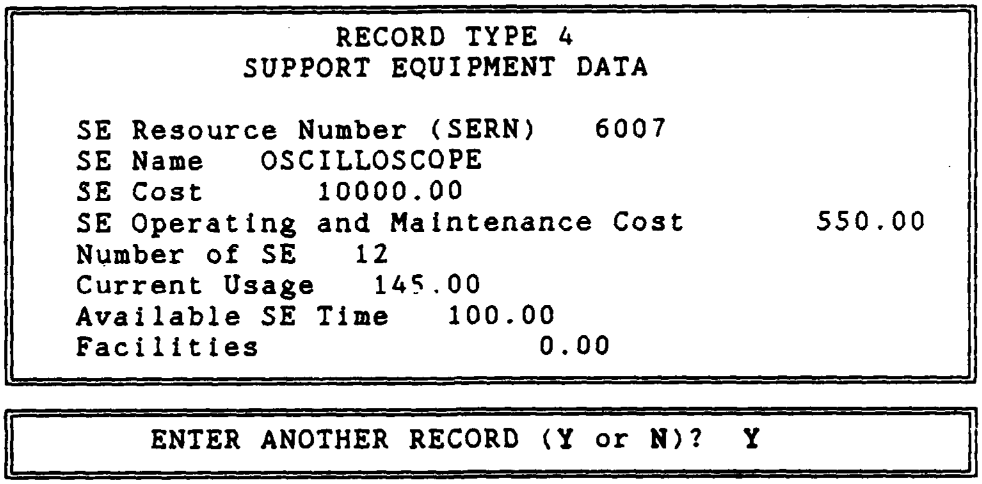
ADD A RECORD
DELETE A RECORD
EDIT A RECORD
RETURN TO EDIT MENU

ENTER **HIGHLIGHTED** SELECTION:

Fig. 9. Edit Choice Menu Screen

The user can enter the highlighted selection associated with the operation desired. The system will only accept entries of <A>, <D>, <E>, or <R>. Any other entry will not be recognized.

Adding Data Records. An entry of <A> at the Edit Choice Menu Screen enables the user to add a new data record within the specific data record type. The add data record screens look similar to the data record display screens shown in Appendix B and perform in an interactive mode to receive data and add a data record to the database file. Following each data record addition the system asks if the user wishes to enter another data record. Upon entry of <Y>, the data previously entered are saved and the fields are cleared. A new data record may now be entered. Upon an entry of <N> the system will return to the Edit Choice Menu. The Support Equipment Add Data Screen is shown in Figure 10 as an example.



```
                  RECORD TYPE 4
             SUPPORT EQUIPMENT DATA

   SE Resource Number (SERN)   6007
   SE Name   OSCILLOSCOPE
   SE Cost      10000.00
   SE Operating and Maintenance Cost       550.00
   Number of SE   12
   Current Usage   145.00
   Available SE Time   100.00
   Facilities                 0.00

         ENTER ANOTHER RECORD (Y or N)?  Y
```

Fig. 10. Support Equipment Add Data Screen

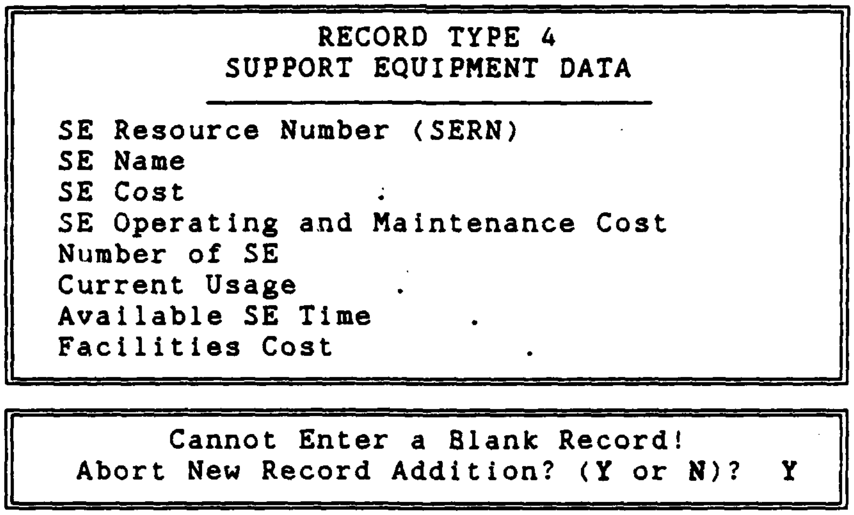
Blank records cannot be entered in the database files. In order to keep the user from entering a blank data record, the system checks a significant data field within the data record types. The significant data fields for each data record type are listed in Table IV.

If the user enters a blank field into one of the significant data fields associated with the data record types shown in Table IV, the system assumes that the user is trying to enter a blank record. The system will notify the user that the entry of a blank record has been attempted and offers the option to abort the addition of this blank data

Table IV. Significant Data Fields

DATA RECORD TYPE	SIGNIFICANT DATA FIELD
Support Equipment Data	SE Resource Number (SERN)
LRU Data	LRU Identifier
LRU Failure Mode Data	LRU Identifier
SRU Data	SRU Identifier
LRU/SRU/SE Cross Reference Data	LRU or SRU Identifier
Individual Factors	LRU Identifier

record. This Blank Record Entry Screen is shown in Figure 11. The system defaults to <Y> at the Abort New Record Addition prompt and the user must enter an <N> in order to save this data record addition. Following this, the user has the option to enter another record as shown in Figure 10 and explained above.

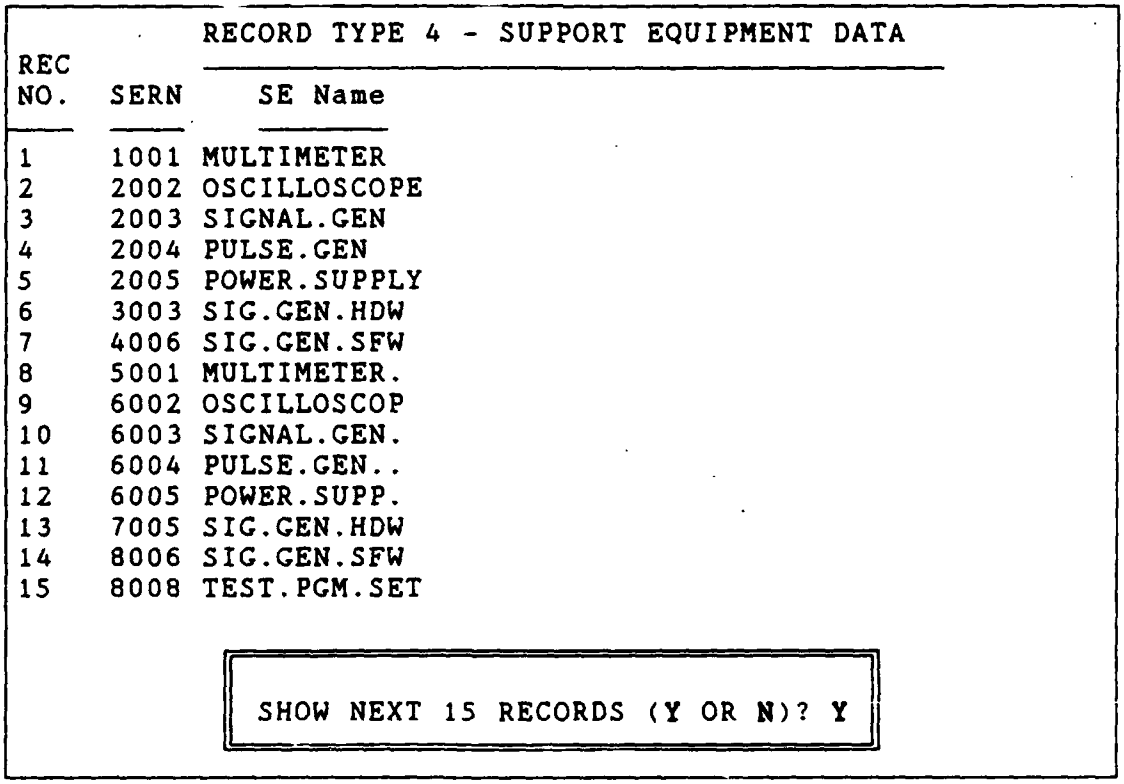


```
                RECORD TYPE 4
            SUPPORT EQUIPMENT DATA

   SE Resource Number (SERN)
   SE Name
   SE Cost
   SE Operating and Maintenance Cost
   Number of SE
   Current Usage
   Available SE Time
   Facilities Cost

          Cannot Enter a Blank Record!
      Abort New Record Addition? (Y or N)?  Y
```

Fig. 11. Blank Record Entry Screen

Deleting Data Records. An entry of <D> at the Edit Choice Menu enables the user to delete data records within the specific data record type. The system will list the data records stored in the database file for the record type selected at the Data Edit Screen. The data records are listed in ascending order of the significant data field. The significant data fields are the same as those listed in Table IV. A sample of each View Data Records Screens is shown in Appendix C. Record numbers may or may not be in order depending on the order in which the data was entered. Record numbers show the order that the records are stored in within the database file.

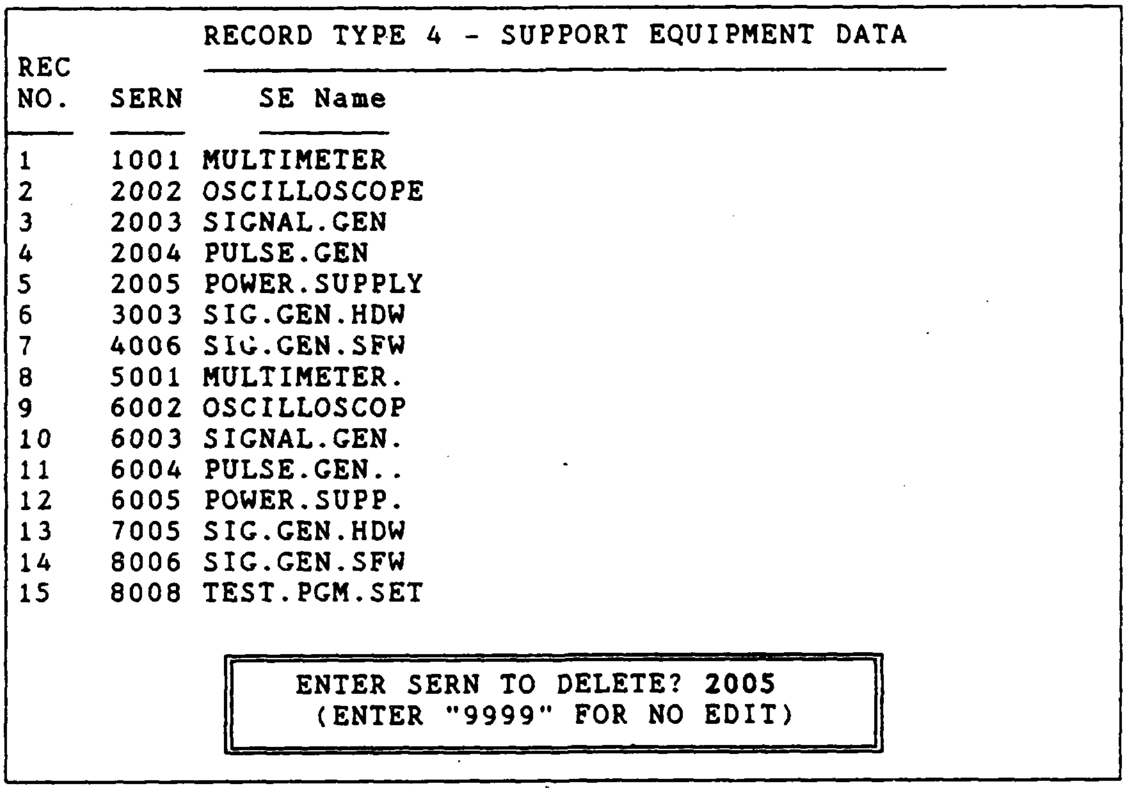
Due to screen size limitations, 15 data records are listed at a time as shown in the View Data Records Screen in Figure 12.



```
              RECORD TYPE 4 - SUPPORT EQUIPMENT DATA
REC
NO.   SERN     SE Name

1     1001 MULTIMETER
2     2002 OSCILLOSCOPE
3     2003 SIGNAL.GEN
4     2004 PULSE.GEN
5     2005 POWER.SUPPLY
6     3003 SIG.GEN.HDW
7     4006 SIG.GEN.SFW
8     5001 MULTIMETER.
9     6002 OSCILLOSCOP
10    6003 SIGNAL.GEN.
11    6004 PULSE.GEN..
12    6005 POWER.SUPP.
13    7005 SIG.GEN.HDW
14    8006 SIG.GEN.SFW
15    8008 TEST.PGM.SET

         SHOW NEXT 15 RECORDS (Y OR N)? Y
```

Fig. 12. View Data Records Screen

The user has the option of listing the next 15 records with the default to the Show Next 15 Records prompt being <Y>. A response of <N> by the user ends the data records listing and the system brings up the Delete Data Records Screen as shown in Figure 13. This screen allows the user to enter the value of the significant field for the data record be to deleted.

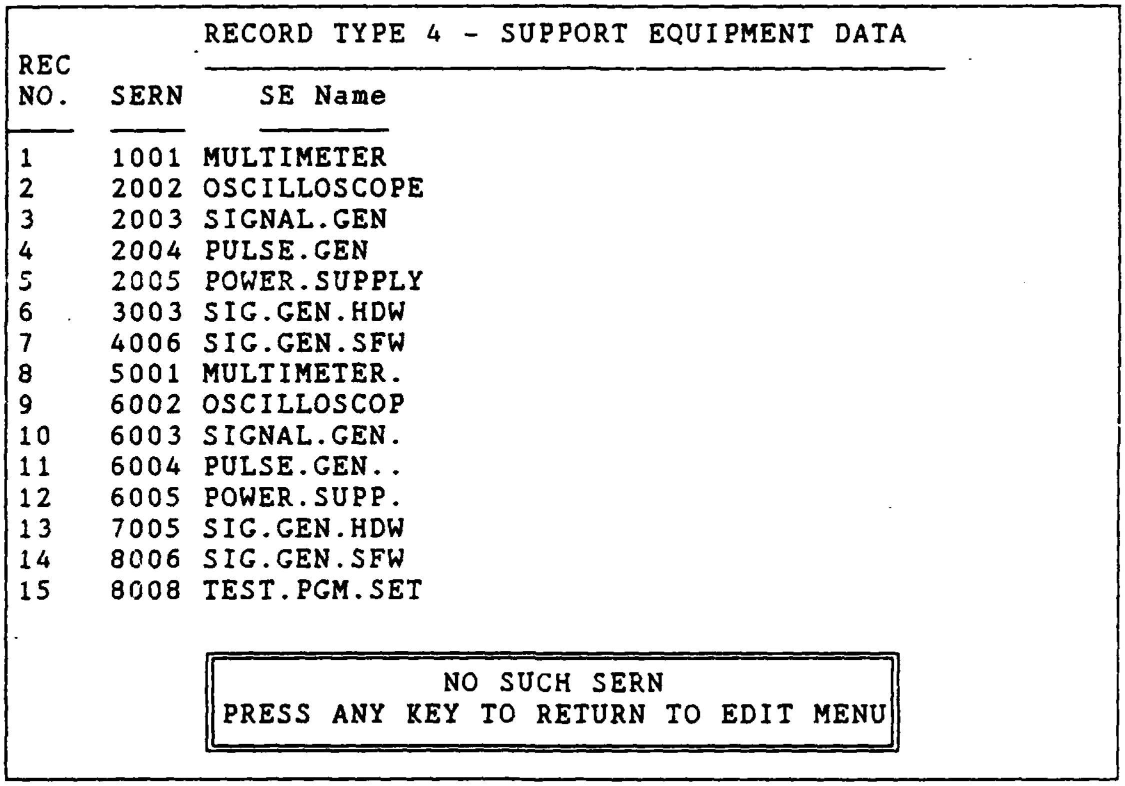


```
            RECORD TYPE 4 - SUPPORT EQUIPMENT DATA
REC
NO.   SERN     SE Name

1     1001 MULTIMETER
2     2002 OSCILLOSCOPE
3     2003 SIGNAL.GEN
4     2004 PULSE.GEN
5     2005 POWER.SUPPLY
6     3003 SIG.GEN.HDW
7     4006 SIG.GEN.SFW
8     5001 MULTIMETER.
9     6002 OSCILLOSCOP
10    6003 SIGNAL.GEN.
11    6004 PULSE.GEN..
12    6005 POWER.SUPP.
13    7005 SIG.GEN.HDW
14    8006 SIG.GEN.SFW
15    8008 TEST.PGM.SET

              ENTER SERN TO DELETE? 2005
               (ENTER "9999" FOR NO EDIT)
```

Fig. 13. Delete Data Records Screen

If no deletion is desired, the user may enter <9999> and the system will return to the Edit Menu with no data record being deleted from the database file.

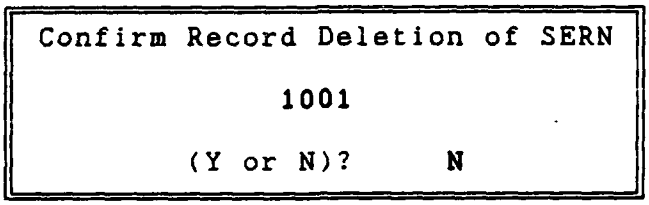
Upon entry of an invalid significant field, that is, one that not stored in the database file of the specified record type, the system informs the user that there is no such data record containing the significant field entered. The system then returns to the Edit Menu. The Invalid Significant Field Screen for support equipment data is shown in Figure 14.



```
              RECORD TYPE 4 - SUPPORT EQUIPMENT DATA
REC
NO.   SERN       SE Name

1     1001 MULTIMETER
2     2002 OSCILLOSCOPE
3     2003 SIGNAL.GEN
4     2004 PULSE.GEN
5     2005 POWER.SUPPLY
6     3003 SIG.GEN.HDW
7     4006 SIG.GEN.SFW
8     5001 MULTIMETER.
9     6002 OSCILLOSCOP
10    6003 SIGNAL.GEN.
11    6004 PULSE.GEN..
12    6005 POWER.SUPP.
13    7005 SIG.GEN.HDW
14    8006 SIG.GEN.SFW
15    8008 TEST.PGM.SET

                          NO SUCH SERN
              PRESS ANY KEY TO RETURN TO EDIT MENU
```

Fig. 14. Invalid Significant Field Screen

Upon entry of a valid significant field, that is, one that is stored in the database file of the specific record type, the user is required to confirm data record deletion. In Figure 15 the Data Record Deletion Confirmation Screen is shown for a support equipment significant field SE resource number of 1001. The default at this deletion confirmation prompt is <N>. An entry of <Y> will delete the data record containing the significant field shown and the system returns to the Edit Choice Menu for the record type chosen at the Data Edit Screen.



```
Confirm Record Deletion of SERN
              1001
        (Y or N)?     N
```

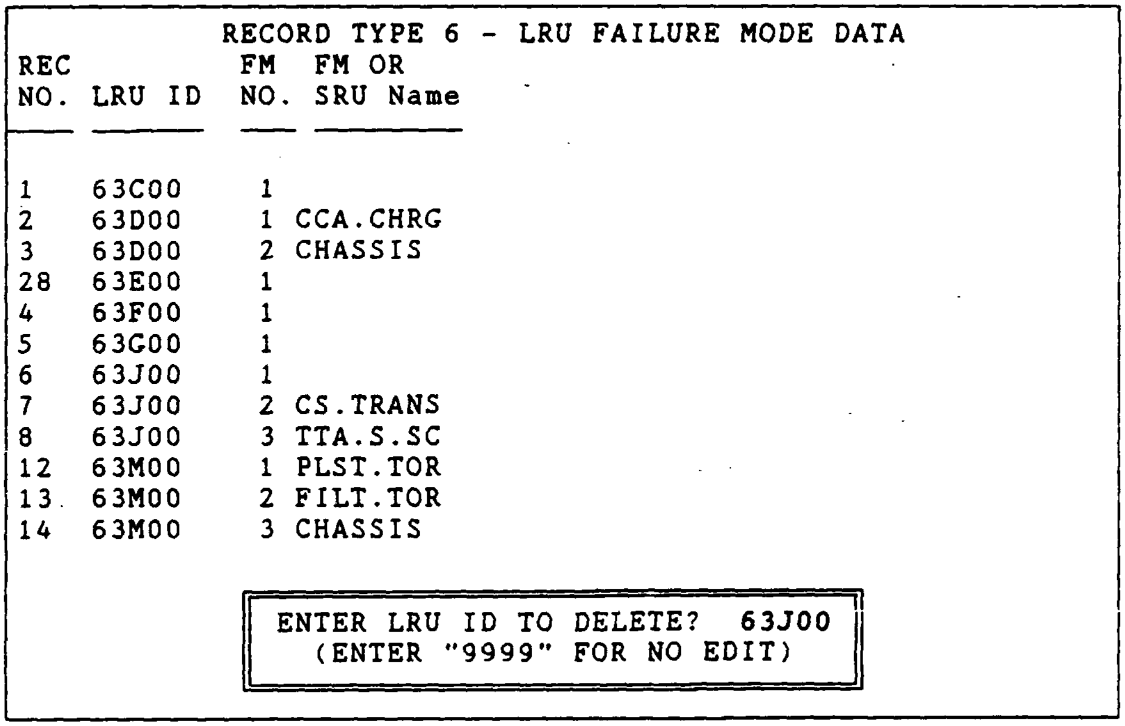
Fig. 15. Data Record Deletion Confirmation Screen

The Delete Data Records Screens work in a similar manner for each of the data record types listed in Table IV. The only exceptions are the LRU Failure Mode and LRU/SRU/SE Cross Reference Data Record Types.

Because of the nature of the data stored in the database files for these data record types, multiple data records stored in the database file may contain the same value for the significant field. A second discriminator distinguishes between data records with identical values in the significant fields in order to be able to identify the correct data record to delete or edit.

For both the LRU Failure Mode Data and the LRU/SRU/SE Cross Reference Data Record Types, this second discriminator is the Failure Mode Number. The Delete Data Records Screen for the LRU Failure Mode Data Record with the Significant

Field prompt is shown in Figure 16. The Delete Data Records Screen for LRU Failure Mode Data Record with the Failure Mode prompt is shown in Figure 17.



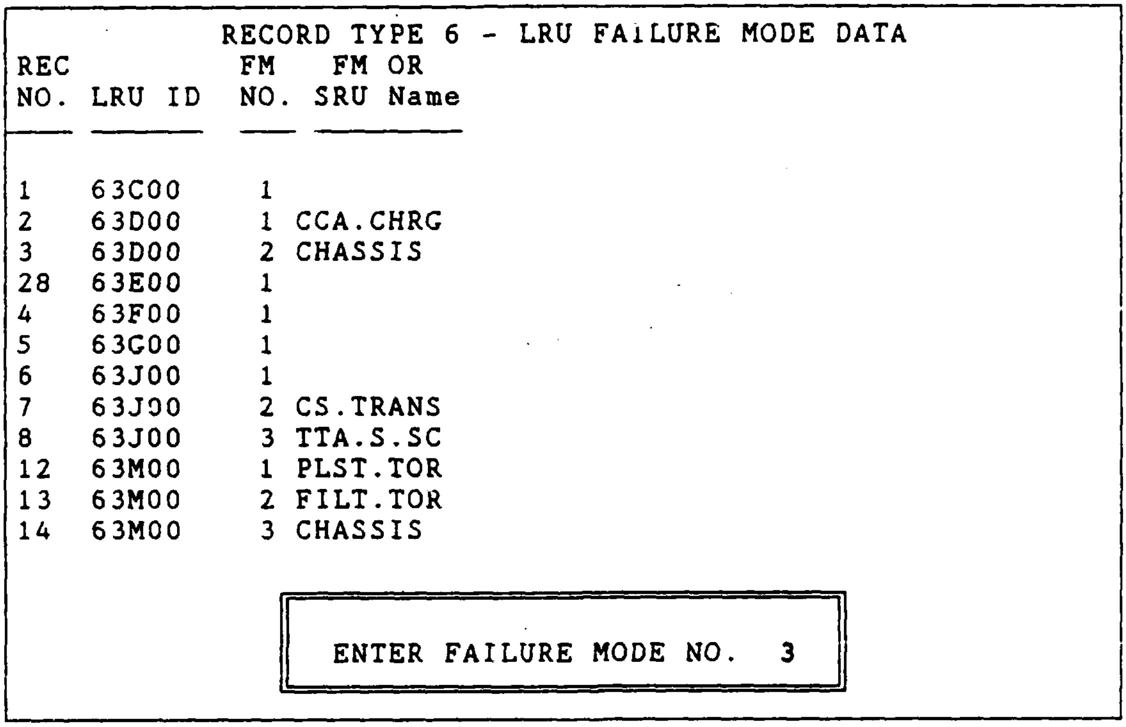
```
                RECORD TYPE 6 - LRU FAILURE MODE DATA
REC                FM   FM OR
NO.  LRU ID        NO.  SRU Name
---  ------        ---  --------

1    63C00          1
2    63D00          1 CCA.CHRG
3    63D00          2 CHASSIS
28   63E00          1
4    63F00          1
5    63G00          1
6    63J00          1
7    63J00          2 CS.TRANS
8    63J00          3 TTA.S.SC
12   63M00          1 PLST.TOR
13   63M00          2 FILT.TOR
14   63M00          3 CHASSIS

                   ENTER LRU ID TO DELETE?  63J00
                     (ENTER "9999" FOR NO EDIT)
```

Fig. 16. Delete Data Records Screen
with Significant Field Prompt

For example, the LRU identifier, 63J00, shown in Figure 16 has three different failure modes. Therefore, 63J00 appears in the database file three times, once for each failure mode (record numbers 6, 7, and 8). In order to delete the third failure mode record for LRU identifier 63J00, the user must enter <63J00> at the ENTER LRU ID TO DELETE prompt as shown in Figure 16. The user must then enter <3> at the ENTER FAILURE MODE NO. prompt as shown in

Figure 17. Following this, the user must confirm the record deletion as described above.

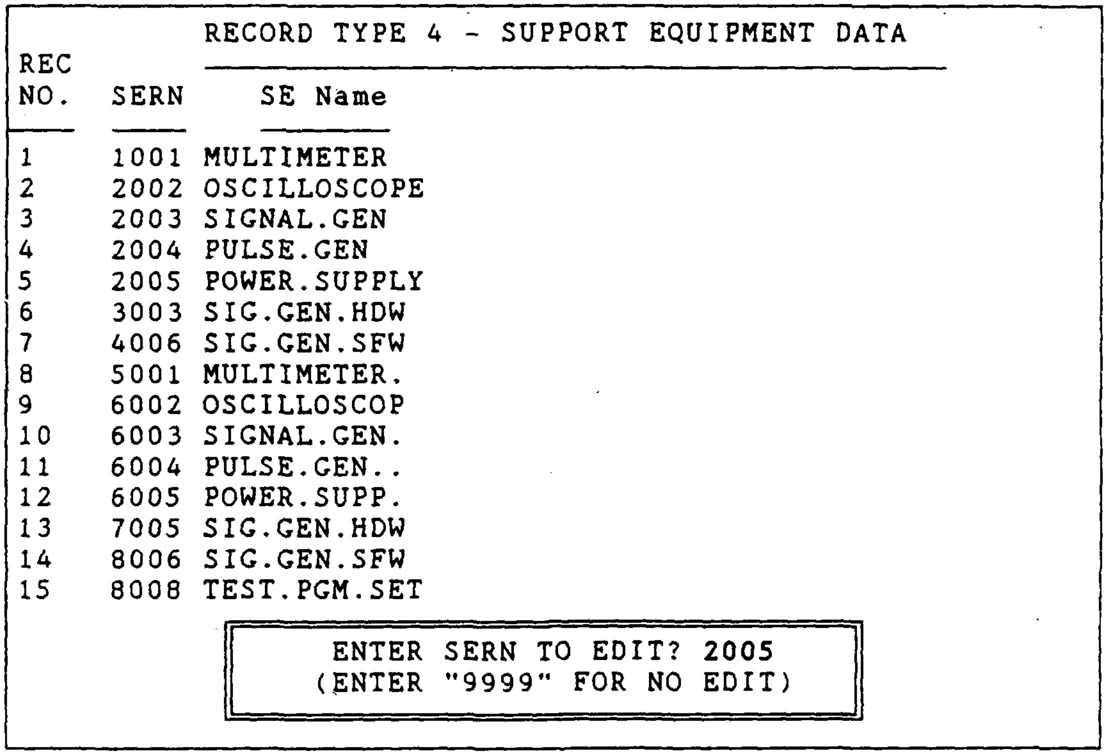
The Edit Data Records Screens for LRU/SRU/SE Cross Reference Data Records are similar to Figures 16 and 17 with different data record type headers, as appropriate.



```
                RECORD TYPE 6 - LRU FAILURE MODE DATA
REC             FM    FM OR
NO.  LRU ID     NO.  SRU Name
---  ------     ---  --------

1    63C00       1
2    63D00       1 CCA.CHRG
3    63D00       2 CHASSIS
28   63E00       1
4    63F00       1
5    63G00       1
6    63J00       1
7    63J00       2 CS.TRANS
8    63J00       3 TTA.S.SC
12   63M00       1 PLST.TOR
13   63M00       2 FILT.TOR
14   63M00       3 CHASSIS

                 ENTER FAILURE MODE NO.  3
```

Fig. 17. Delete Data Records Screen
with Failure Mode Prompt

Editing Data Records. An entry of <E> at the Edit Choice Menu enables the user to edit data records within the specific data record type. The system will list the data records stored in the database file for that record type in the same manner as the View Data Records Screen shown in Figure 12. Similar to deleting data, the user can list the data records by significant field and enter the significant field of the data record he wants to edit. The Edit Data Records Screen for the Support Equipment Data Record type is shown in Figure 18.



```
                RECORD TYPE 4 - SUPPORT EQUIPMENT DATA
REC
NO.    SERN      SE Name

1      1001 MULTIMETER
2      2002 OSCILLOSCOPE
3      2003 SIGNAL.GEN
4      2004 PULSE.GEN
5      2005 POWER.SUPPLY
6      3003 SIG.GEN.HDW
7      4006 SIG.GEN.SFW
8      5001 MULTIMETER.
9      6002 OSCILLOSCOP
10     6003 SIGNAL.GEN.
11     6004 PULSE.GEN..
12     6005 POWER.SUPP.
13     7005 SIG.GEN.HDW
14     8006 SIG.GEN.SFW
15     8008 TEST.PGM.SET

                 ENTER SERN TO EDIT? 2005
                (ENTER "9999" FOR NO EDIT)
```

Fig. 18. Edit Data Records Screen

The system performs the same check for a valid significant field as when deleting data records. The system displays the same screen shown in Figure 14 if an invalid significant field is entered.

When a valid significant field is entered, the system displays a screen similar to the Display Data Records Screens shown in Appendix B. These screens are interactive and allow the user to change data in any of the fields within the data record. In order to edit, the user can press either <return> or <down arrow> to place the cursor on the field to edit. Upon entering <return> after editing a field, the edited data is stored in the database file.

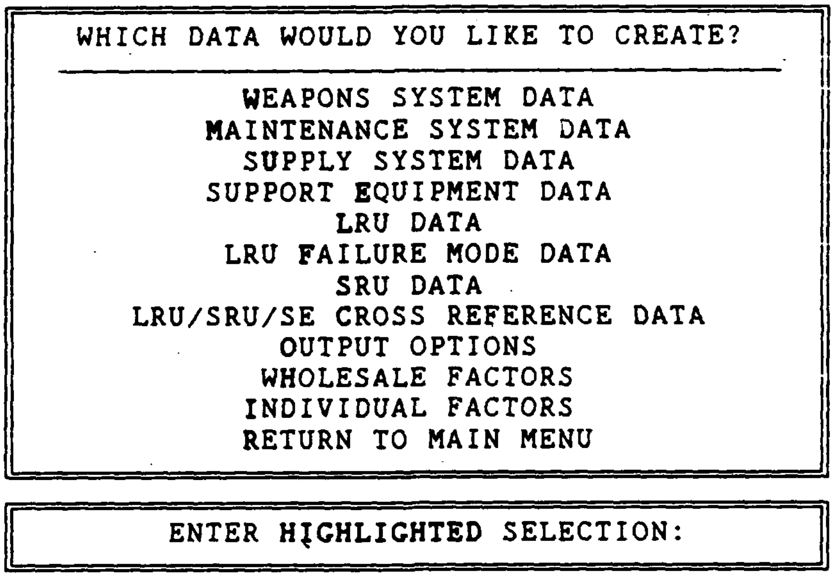
When the edit is complete the system returns to the Edit Choice Menu. Should the user choose not to edit data contained in these records, he may step through the fields on the screen by pressing the <return> key. Pressing <pg dn> will exit the data record screen without any further changes to the data in the database file not previously edited. The system then returns to the Edit Choice Menu.

The Edit Data Records Screens work in a similar fashion for each of the data record types listed in Table IV. As with the Delete Data Records Screens, the only exceptions are the LRU Failure Mode Data and the LRU/SRU/SE Cross Reference Data Record Types. The same process to distinguish between data records with identical values in the significant fields using a second discriminator is followed.

Creating a New File

A selection of <C> at the Main Menu selection prompt displays the Create Data Records Screen shown in Figure 19. In this module the user can create new database files containing new data for any or all data record types. In order to do so the system deletes the data currently in the database file for the record type selected and allows the user to enter data records into a blank database file. It essentially wipes the slate clean in order to enter data for a different weapons system.

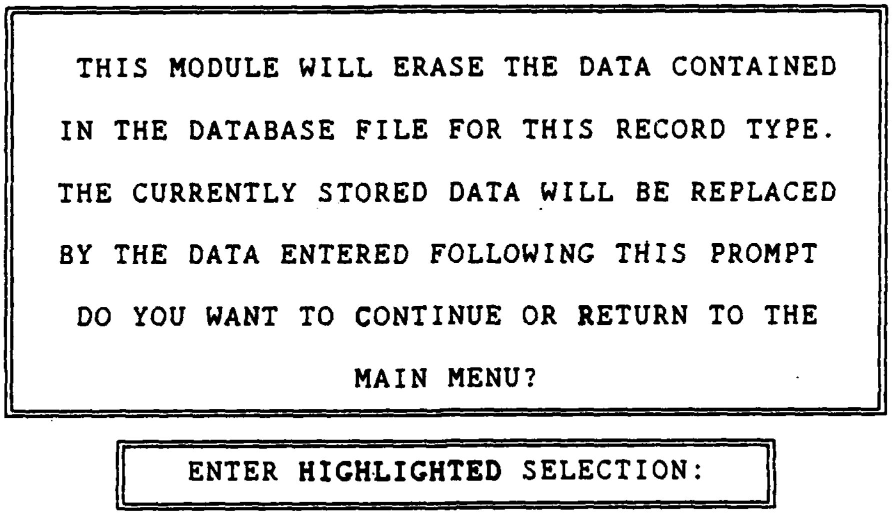
To ensure that the user desires to create a new database file for the record type selected, a warning and confirmation screen is displayed and the user must confirm continuation of the creation module.



```
WHICH DATA WOULD YOU LIKE TO CREATE?

WEAPONS SYSTEM DATA
MAINTENANCE SYSTEM DATA
SUPPLY SYSTEM DATA
SUPPORT EQUIPMENT DATA
LRU DATA
LRU FAILURE MODE DATA
SRU DATA
LRU/SRU/SE CROSS REFERENCE DATA
OUTPUT OPTIONS
WHOLESALE FACTORS
INDIVIDUAL FACTORS
RETURN TO MAIN MENU

ENTER HIGHLIGHTED SELECTION:
```

Fig. 19. Create Data Records Screen

The Creation Confirmation Screen is shown in Figure 20. If the user does not want to erase and create a new database file for the selected record type, a response of <R> will return the system to the Main Menu. Upon entering a <C> to continue the creation module the system will delete the records stored in the database files for the specified data record type and will display the same screens as that displayed for the Add Data Records Screens in order to create/enter new data records into the database files. Similar to the Add Data Records Screens the system will not allow blank records to be entered.



THIS MODULE WILL ERASE THE DATA CONTAINED
IN THE DATABASE FILE FOR THIS RECORD TYPE.
THE CURRENTLY STORED DATA WILL BE REPLACED
BY THE DATA ENTERED FOLLOWING THIS PROMPT
DO YOU WANT TO CONTINUE OR RETURN TO THE
MAIN MENU?

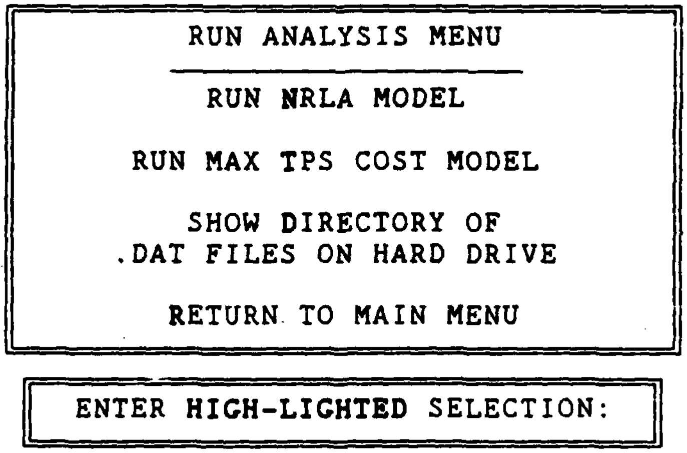
ENTER **HIGHLIGHTED** SELECTION:

Fig. 20. Creation Confirmation Screen

Running the Analysis Models

A selection of <R> at the Main Menu selection prompt displays the Run Analysis Menu as shown in Figure 21. This module enables the user to run the analysis models portion of the system.

At this menu the user has the option to run either the NRLA model or the TPS Max Cost model. The user may also display the directory of .DAT files stored on the hard drive or return to the Main Menu. At this prompt the system only accepts entries <N>, <T>, <D>, or <R>. Any other entry is not recognized by the system.



RUN ANALYSIS MENU

RUN NRLA MODEL

RUN MAX TPS COST MODEL

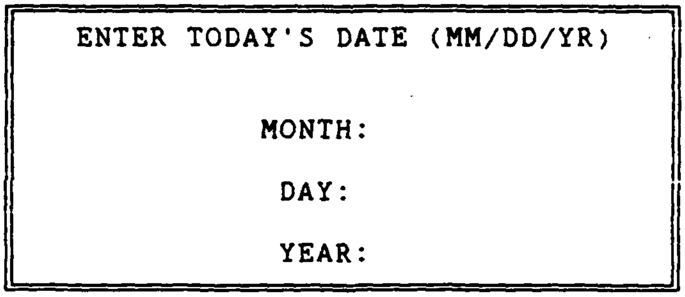
SHOW DIRECTORY OF
.DAT FILES ON HARD DRIVE

RETURN TO MAIN MENU

ENTER HIGH-LIGHTED SELECTION:

Fig. 21. Run Analysis Menu

Running the NRLA Model. In order to run the NRLA model, the user selects <N> at the Run Analysis Menu selection prompt and the system displays the Date Entry Screen shown in Figure 22.



ENTER TODAY'S DATE (MM/DD/YR)

MONTH:

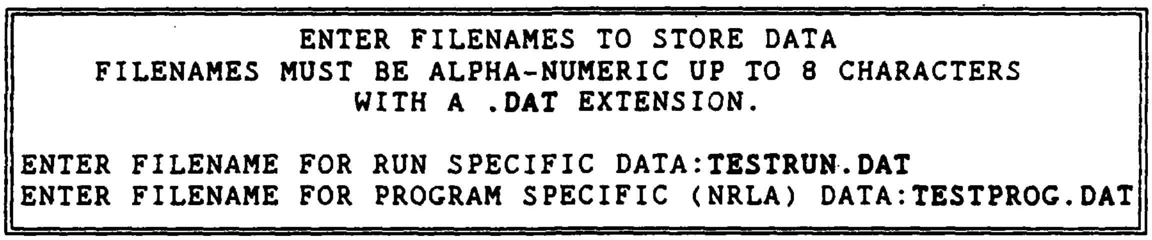
DAY:

YEAR:

Fig. 22. Date Entry Screen

Following entry of the date, the system displays the Data Filename Storage Screen as shown in Figure 23. The user must enter the filenames to which the data stored in the database files will be written. NRLA will use the data written to these files for its analysis. The filenames

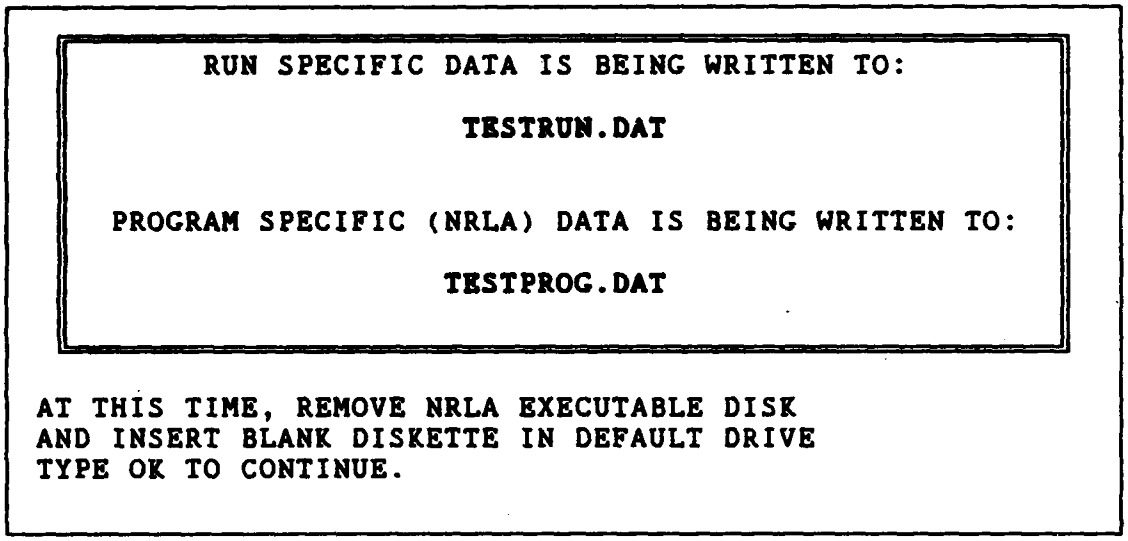
TESTRUN.DAT and TESTPROG.DAT shown in Figure 23 are sample filenames. Any filename up to eight characters long with a .DAT extension may be entered. The .DAT extension is required for the NRLA model to process the data and must be entered as part of both filenames.



```
                   ENTER FILENAMES TO STORE DATA
     FILENAMES MUST BE ALPHA-NUMERIC UP TO 8 CHARACTERS
                     WITH A .DAT EXTENSION.

ENTER FILENAME FOR RUN SPECIFIC DATA:TESTRUN.DAT
ENTER FILENAME FOR PROGRAM SPECIFIC (NRLA) DATA:TESTPROG.DAT
```

Fig. 23. Data Filename Storage Screen

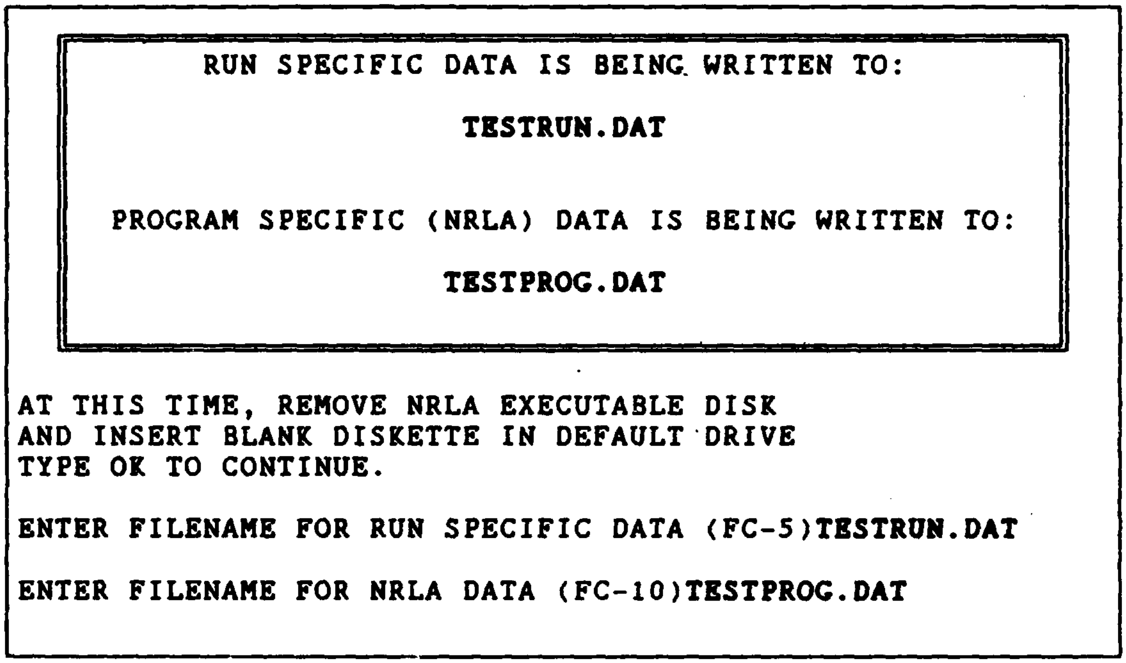
Upon entry of the filenames the system writes the data to the specified files and, as the data is being written, displays the screen showing those filenames. The system then runs the NRLA model as shown in the Data Writing Screen in Figure 24. At this point the NRLA model is enacted and the user should press the <control> and <P> keys simultaneously in order to send the output to the printer. If no printout is desired, the user enters <OK> to continue the NRLA model. A blank diskette need not be inserted in the default as specified. The user need only enter the filenames for the Run Specific Data and the Program Specific Data including the .DAT extension as shown in Figure 25. Upon entry of the data filenames the NRLA model will run.



```
           RUN SPECIFIC DATA IS BEING WRITTEN TO:

                        TESTRUN.DAT


     PROGRAM SPECIFIC (NRLA) DATA IS BEING WRITTEN TO:

                       TESTPROG.DAT

AT THIS TIME, REMOVE NRLA EXECUTABLE DISK
AND INSERT BLANK DISKETTE IN DEFAULT DRIVE
TYPE OK TO CONTINUE.
```

Fig. 24. Data Writing Screen

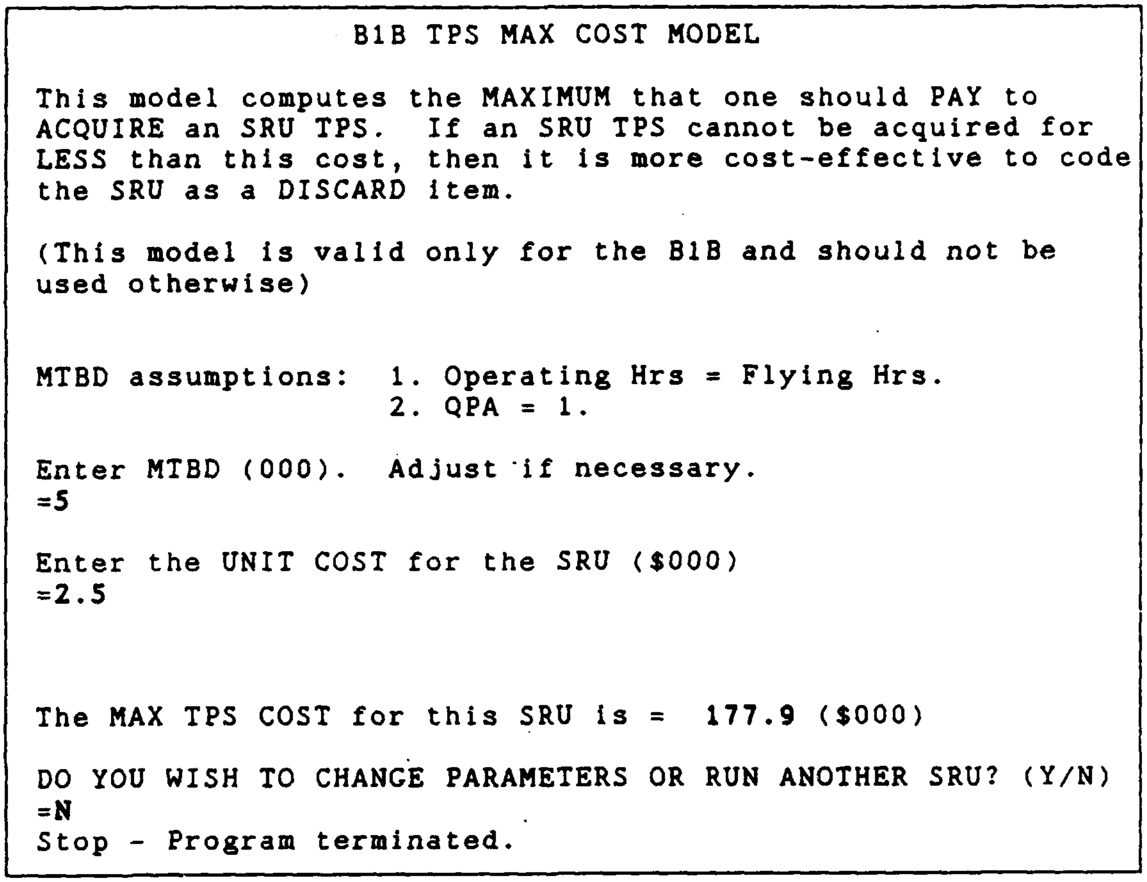
To abort the NRLA analysis, the user can press the <control> and <C> keys simultaneously. The analysis will stop and the system will return to the Main Menu.



```
           RUN SPECIFIC DATA IS BEING WRITTEN TO:

                        TESTRUN.DAT


     PROGRAM SPECIFIC (NRLA) DATA IS BEING WRITTEN TO:

                       TESTPROG.DAT

AT THIS TIME, REMOVE NRLA EXECUTABLE DISK
AND INSERT BLANK DISKETTE IN DEFAULT DRIVE
TYPE OK TO CONTINUE.

ENTER FILENAME FOR RUN SPECIFIC DATA (FC-5)TESTRUN.DAT

ENTER FILENAME FOR NRLA DATA (FC-10)TESTPROG.DAT
```

Fig. 25. NRLA Datafile Run Screen

Running the TPS Max Cost Model. A selection of <T> at the Run Analysis Menu selection prompt will run the TPS Max Cost model. This model requires that the LRU/SRU mean time between demand (MTBD) as well as the unit cost of the LRU/SRU be entered. The model will then return the maximum cost that the government should pay in acquiring a TPS. This is shown in Figure 26.



```
                        B1B TPS MAX COST MODEL

This model computes the MAXIMUM that one should PAY to
ACQUIRE an SRU TPS.  If an SRU TPS cannot be acquired for
LESS than this cost, then it is more cost-effective to code
the SRU as a DISCARD item.

(This model is valid only for the B1B and should not be
used otherwise)


MTBD assumptions:  1. Operating Hrs = Flying Hrs.
                   2. QPA = 1.

Enter MTBD (000).  Adjust if necessary.
=5

Enter the UNIT COST for the SRU ($000)
=2.5



The MAX TPS COST for this SRU is =  177.9 ($000)

DO YOU WISH TO CHANGE PARAMETERS OR RUN ANOTHER SRU? (Y/N)
=N
Stop - Program terminated.
```

Fig. 26. TPS Max Cost Model

Viewing .DAT Files. An entry of <D> at the Run Analysis Menu selection prompt allows the user to view the list of filenames that are saved on the hard drive with the .DAT extension. This permits the user to see what has been saved with a .DAT extension should he desire to write data to files with different filenames. By naming the filename with a .DAT extension, the user can view easily what has been saved.

Saving Data for Different Weapons Systems

Should the user desire to use NRLA INTERFACE ~d this DSS to create, edit and store data for several weapons systems, the data for each weapons system should be stored in separate subdirectories on a five and one-quarter inch floppy diskette.

The 13 database files shown in Appendix A contain the structure of the data as well as the actual data itself pertaining to the weapons system under analysis. When performing analysis on several weapons systems, these 13 database files should be carefully stored so as to not write data for one weapons system over the data of for another weapons system. For example, if the user has been running the NRLA model for Weapons System A, the database files on the hard drive contain all the relevant data for that weapons system. If the user then desires to run the NRLA model for Weapons System B and not lose the data for Weapons System A, he must follow these steps:

1) At the NRLA INTERFACE Main Menu, enter <X> to exit to the computer's disk operating system.

2) Insert a blank diskette into drive A:.

3) Make a subdirectory on the drive A: diskette for storing the database files containing Weapons Systems A data.

4) Change to this drive A: subdirectory and type <COPY C: *.DBF A:>

Each of the 13 database files shown in Appendix A will be copied to this drive A: subdirectory. The user may now enter the DSS and run the create module to store data for Weapons System B. The above steps should be followed to save Weapons System B data to a floppy diskette making sure the database files (i.e., those with .DBF extensions) are stored in a different subdirectory than those stored for Weapons System A.

Once the database files for Weapons System B are stored on a floppy diskette, the data for Weapon System A can be copied back to the hard drive for further data manipulation and analysis. This is done from the disk operating system by entering the correct directory on the drive A: diskette and typing <COPY A: *.DBF C:>. This will overwrite the 13 .DBF files on the hard drive with those contained on the drive A: diskette. This data may now be manipulated using NRLA INTERFACE.

V. Conclusions and Recommendations

Conclusions

This thesis began by investigating the applicability of the development of a specific decision support system for enhancing the use of NRLA in the acquisition of test program sets. In order to determine the applicability of this DSS development, many research questions were presented in Chapter I. In the subsequent chapters of this thesis these questions were investigated and answered. A summary of the answers to each research question is provided here.

Research Question #1

> 1) How can the development of a DSS using NRLA as the models base enhance the repair level analysis process?

The development of a DSS enhanced the repair level analysis process by providing a much needed user interface and database management system facility that greatly improves the entering and editing of data used by the NRLA model. The applications programs developed here remove the burden of having to manipulate unformatted input data files. The development of applications programs that make up the user interface and DBMS facilitate the iterative accomplishment of the NRLA model which is required throughout the weapons system development process. These programs ease the iterative data manipulation process that is necessary when weapons system parameters change.

With the incorporation of a model that calculates TPS breakeven costs, an additional decision-making capability is added to the repair level decision process. This model innovatively uses the NRLA model output to provide cost guidelines used in the acquisition of TPSs.

Research Question #2

> 2) What similar software systems are being used elsewhere in the Air Force to meet similar requirements? If software systems/modules are available, can they be adapted for use here?

There was no evidence found of a similar software system being used in the Air Force to meet the needs of the user interface and DBMS facility for use in this development process. However, dBase III Plus$^{(R)}$ is a software package that is widely used throughout the Air Force and was investigated for use here.

A TPS Max Cost model developed for the B-1B program was available and incorporated for use in this DSS. It provides cost guidelines for the acquisition of SRU test program sets.

Research Question #3

> 3) If there are no adaptable systems available, could a system be developed that will make use of the existing model?

dBase III Plus$^{(R)}$ was chosen as the software package to be used for development of this DSS. The applications programs written here greatly enhance the use of the existing NRLA model. They incorporate the TPS Max Cost model as well as provide a totally integrated system made up of a user interface, a database, a database management system, an analysis models base and an output generator.

Research Question #4

4) If writing a procedure for the modules,

> a) what are the specific NRLA compatibility requirements for the system to work successfully?

The specific NRLA compatibility requirements are outlined in Chapter 2 of the *Network Repair Level Analysis Model User's Guide* (3:5-57). These requirements include the specifications for the data fields that are required for the program specific and the run specific input data files. The *Network Repair Level Analysis User's Guide* describes in detail how these input data files should be formatted in order that the NRLA model will properly run.

> b) which commercial software packages will best satisfy these requirements?

dBase III Plus(R) was chosen as the software package used in this research effort because of its capability to integrate applications programs with its database and database management system facilities. dBase III Plus(R) allows program control to pass back and forth from the applications programs to models external to the dBase III environment. dBase III Plus(R) is also widely used throughout the Air Force.

Research Question #5

5) What criteria and concerns must be considered to ensure successful acceptance and implementation by the users?

In order for this system to be successfully accepted and implemented by the users, their needs had to be identified. Unstructured interviews and technical interchange meetings were held throughout the development process with personnel from the NRLA staff office (AFALC/LSX), maintenance planners and acquisition managers from ASD at WPAFB.

The criteria and concerns that were considered could be separated into two categories: development of a user interface and DBMS to enhance the usefulness of the NRLA model; and the integration of a model to provide cost guidelines in the acquisition of test program sets. The applications programs written in dBase III Plus(R) fulfilled the required criteria considerations and the integration of the TPS Max Cost model provided an effective initial specific DSS for the B-1B program. This is, however, the only program for which this model should be used for, since it was developed using B-1B system data.

The initial specific DSS that was developed can be stored on one five and one quarter inch floppy diskette. This system can be run on any IBM(R) compatible microcomputer with 640 kilobytes of random access memory (RAM), a 20 megabyte hard drive, a monitor, and a 120 column printer. This system should be used in conjunction with the existing

NRLA documentation as well as with the user's reference guide provided here.

Recommendations for Future Research

Validation. Since the system developed here is indeed an initial specific DSS, it is recommended that the validation process be continued to ensure that the system will perform as intended under all necessary conditions. This would consist of extensive testing using real or sample data and comparing the output results generated by the NRLA model using the DSS to input the data and run the model to output generated by the NRLA model not using the DSS to input the data. This testing should be continued until all the combinations of data input possibilities have been exhausted and the system performs as required. This will ensure that the applications programs written to make up the DBMS are properly manipulating the data so that the NRLA model can read them correctly.

TPS Max Cost Model Enhancement. Since the TPS Max Cost model was developed using data pertaining to the B-1B weapons system, it should not be used for other weapons systems. This model should be used as a basis for developing a generic model that could be further integrated with the NRLA model output and run to directly draw from this data and provide TPS cost guidelines for the system under analysis. This development had been envisioned as part of this research; however, due to time constraints, it could not be completed.

Completion of this enhancement would provide a true DSS for all weapons systems.

LSA Integration. Another potential enhancement could be to integrate/interface this system with the Logistics Support Analysis (LSA) data base in order to provide the latest up-to-date component reliability and related logistics factors. This will provide "real-time" access to the required data as design changes are made without having to manually enter certain data.

Expert System Integration. A final potential enhancement would be to integrate this system with an expert system that considers other non-economic factors such as vulnerability, mobility, deployment, and safety. This would provide a system that would arrive at the best repair level decision by taking into account both economic and non-economic factors.

Appendix A: Database File Structures

```
Structure for database: C:weapon_s.dbf
Number of data records:       1
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  END_ITEM    Character     12
    2  NO_I_LOC    Numeric        3
    3  RATIO_OS    Numeric        4      2
    4  YRS_SYS_LI  Numeric        2
    5  EQ_WS_ILOC  Numeric        4
    6  MO_OPS_HRS  Numeric        3
    7  SE_DEV_CST  Numeric        7
    8  SENSI_ALT   Numeric        1
    9  L_SEN_RANG  Numeric        4      2
   10  U_SEN_RANG  Numeric        4      2
   11  SENSI_TYPE  Numeric        1
   12  OPTION_INF  Character     20
** Total **                      66
```

```
Structure for database: C:maint_sy.dbf
Number of data records:       1
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  INT_MNHRS   Numeric        3
    2  INT_LAB_RT  Numeric        5      2
    3  DEP_MNHRS   Numeric        3
    4  DEP_LAB_RT  Numeric        5      2
    5  INT_TRN_RT  Numeric        5      3
    6  DEP_TRN_RT  Numeric        5      3
** Total **                      27
```

```
Structure for database: C:supply_s.dbf
Number of data records:       1
Date of last update   : 08/23/88
Field  Field Name  Type        Width    Dec
    1  INI_MGT_CS  Numeric         7      2
    2  REC_MGT_CS  Numeric         7      2
    3  BS_MGT_CS   Numeric         7      3
    4  CONUS_SHIP  Numeric         5      3
    5  OS_SHIP     Numeric         5      3
    6  CONUS_PK_C  Numeric         7      3
    7  OS_PK_CST   Numeric         7      3
    8  CONUS_PKWT  Numeric         7      3
    9  OS_PKWT_RT  Numeric         7      3
   10  CONUS_SHRT  Numeric         7      3
   11  OS_SHIP_RT  Numeric         7      3
   12  TEC_DAT_CS  Numeric         6      2
** Total **                       80
```

```
Structure for database: C:se_data.dbf
Number of data records:       0
Date of last update   : 08/23/88
Field  Field Name  Type        Width    Dec
    1  SERN        Numeric         4
    2  SE_NAME     Character      12
    3  SE_COST     Numeric        11      2
    4  SE_O_M_CST  Numeric         8      2
    5  NO_OF_SE    Numeric         2
    6  CURR_USAGE  Numeric         6      2
    7  AVAIL_TIME  Numeric         6      2
    8  FACIL_COST  Numeric        11      2
** Total **                       61
```

```
Structure for database: C:sesort.dbf
Number of data records:       14
Date of last update   : 08/09/88
Field  Field Name  Type       Width    Dec
    1  SERN        Numeric        4
    2  SE_NAME     Character     12
    3  SE_COST     Numeric       11      2
    4  SE_O_M_CST  Numeric        8      2
    5  NO_OF_SE    Numeric        2
    6  CURR_USAGE  Numeric        6      2
    7  AVAIL_TIME  Numeric        6      2
    8  FACIL_COST  Numeric       11      2
** Total **                      61
```

```
Structure for database: C:lru_data.dbf
Number of data records:        0
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  LRU_ID      Character      7
    2  LRU_NAME    Character     12
    3  NO_LRU_END  Numeric        4      1
    4  UNIT_COST   Numeric       10      2
    5  WEIGHT      Numeric        6      1
    6  OPS_RATIO   Numeric        7      3
    7  DEP_RCT_CO  Numeric        7      3
    8  DEP_RCT_OS  Numeric        7      3
    9  INT_RCT_CO  Numeric        7      3
   10  R_FAIL_RIP  Numeric        5      2
   11  NO_GPSE_RQ  Numeric        3
   12  MTBF        Numeric        9      2
** Total **                      85
```

```
Structure for database: C:lru_fm_d.dbf
Number of data records:      33
Date of last update   : 08/01/88
Field  Field Name  Type       Width    Dec
    1  LRU_ID      Character      7
    2  FM_IDEN_NO  Numeric        2
    3  FM_RATIO    Numeric        5      3
    4  SRU_ID      Character      7
    5  FM_SRU_NAM  Character      8
    6  NO_NEW_PRT  Numeric        3
    7  NO_STD_PRT  Numeric        3
    8  REP_PRT_CS  Numeric        8      2
    9  WT_PC_PART  Numeric        7      2
   10  NO_TRN_DEP  Numeric        3
   11  NO_TRN_INT  Numeric        3
   12  DEP_MX_MHR  Numeric        7      3
   13  INT_MX_MHR  Numeric        7      3
   14  WKS_MX_TRN  Numeric        7      3
   15  TRNG_CST    Numeric        8      2
   16  TEC_DT_PGS  Numeric        5      1
   17  NO_SPSE_RQ  Numeric        3
   18  FRC_LRU_FM  Numeric        3
   19  SE_HRS_REP  Numeric        7      3
** Total **                     104

Structure for database: C:sru_data.dbf
Number of data records:      10
Date of last update   : 08/02/88
Field  Field Name  Type       Width    Dec
    1  SRU_ID      Character      7
    2  SRU_CST     Numeric        8      2
    3  SRU_WEIGHT  Numeric        6      1
    4  CST_PP_REP  Numeric        8      2
    5  WT_PP_ASSY  Numeric        7      2
    6  NO_NEW_PP   Numeric        3
    7  NO_STD_PP   Numeric        3
    8  NO_PGS_T_D  Numeric        5      1
    9  NO_KIND_SE  Numeric        2
   10  DEP_RCT_CO  Numeric        7      3
   11  DEP_RCT_OS  Numeric        7      3
   12  INT_RCT     Numeric        7      3
   13  DEP_MX_MHR  Numeric        7      3
   14  INT_MX_MHR  Numeric        7      3
   15  DEP_TRNEES  Numeric        3
   16  INT_TRNEES  Numeric        3
   17  MX_TRN_WKS  Numeric        7      3
   18  MX_TRN_CST  Numeric        8      2
   19  FRC_SRU_DE  Numeric        3
   20  SE_HRS_REP  Numeric        7      3
** Total **                     116
```

```
Structure for database: C:cross_re.dbf
Number of data records:       0
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  REC_TYP_ID  Numeric        1
    2  LRU_SRU_ID  Character      7
    3  FM_ID_NO    Numeric        2
    4  SE_RES_NO1  Numeric        4
    5  SE_RES_NO2  Numeric        4
    6  SE_RES_NO3  Numeric        4
    7  SE_RES_NO4  Numeric        4
    8  SE_RES_NO5  Numeric        4
    9  SE_RES_NO6  Numeric        4
   10  SE_RES_NO7  Numeric        4
   11  SE_RES_NO8  Numeric        4
   12  SE_RES_NO9  Numeric        4
   13  SE_RES_N10  Numeric        4
   14  SE_RES_N11  Numeric        4
   15  SE_RES_N12  Numeric        4
   16  SE_RES_N13  Numeric        4
   17  SE_RES_N14  Numeric        4
   18  SE_RES_N15  Numeric        4
   19  SE_RES_N16  Numeric        4
   20  CONTIN_IND  Numeric        2
** Total **                      77
```

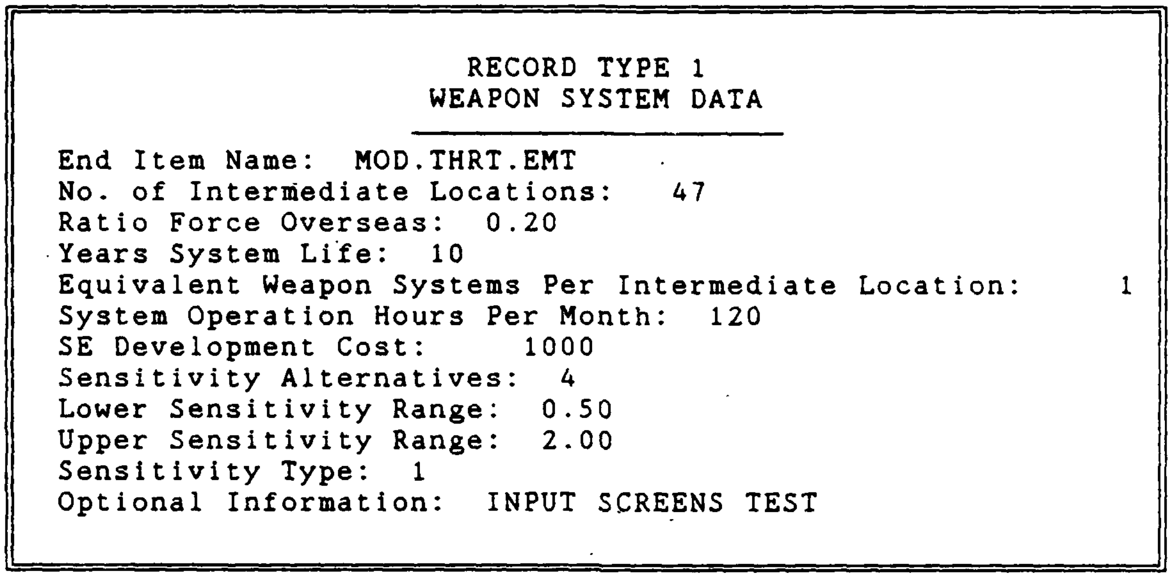
```
Structure for database: C:outp_opt.dbf
Number of data records:       1
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  INPUT_ECHO  Numeric        1
    2  SE_INPUT    Numeric        1
    3  SE_CRS_REF  Numeric        1
    4  SE_CSTS     Numeric        1
    5  SE_RQMTS    Numeric        1
    6  RL_DEC_DET  Numeric        1
    7  LRU_RL_SUM  Numeric        1
    8  OUTP_UNITS  Numeric        1
** Total **                       9
```

```
Structure for database: C:wc_ex_fc.dbf
Number of data records:       1
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  LRU_SRU_CS  Numeric        8      3
    2  LRU_MTBFS   Numeric        8      3
    3  SE_CSTS     Numeric        8      3
    4  X_LRU_DREP  Numeric        1
    5  X_LRU_SCRP  Numeric        1
    6  X_LRU_IREP  Numeric        1
    7  X_SRU_DREP  Numeric        1
    8  X_SRU_SCRP  Numeric        1
    9  X_SRU_IREP  Numeric        1
** Total **                      31
```

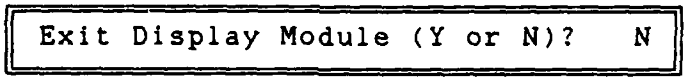
```
Structure for database: C:ind_cnge.dbf
Number of data records:       0
Date of last update   : 08/23/88
Field  Field Name  Type       Width    Dec
    1  SE_RES_1    Numeric        4
    2  SE_CH_FAC1  Numeric        7      2
    3  SE_RES_2    Numeric        4
    4  SE_CH_FAC2  Numeric        7      2
    5  LRU_ID      Character      7
    6  LRU_CST_FA  Numeric        7      2
    7  MTBF_FAC    Numeric        7      2
** Total **                      44
```

```
Structure for database: C:date.dbf
Number of data records:       1
Date of last update   : 08/09/88
Field  Field Name  Type       Width    Dec
    1  MONTH       Numeric        2
    2  DAY         Numeric        2
    3  YEAR        Numeric        2
** Total **                       7
```

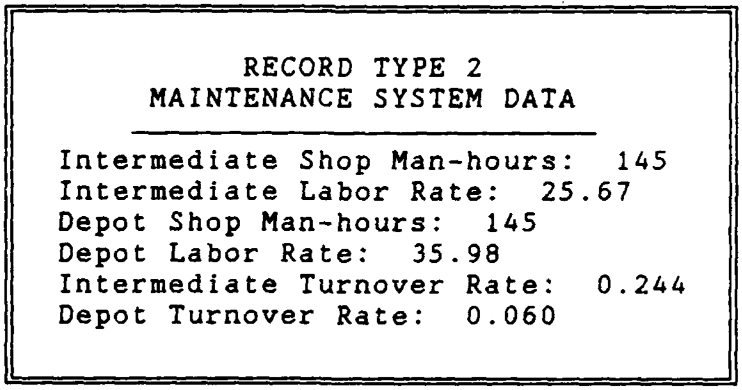
Appendix B: Data Records Display Screens



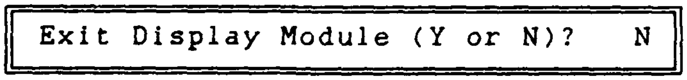
```
                      RECORD TYPE 1
                   WEAPON SYSTEM DATA
   End Item Name:  MOD.THRT.EMT
   No. of Intermediate Locations:   47
   Ratio Force Overseas:  0.20
   Years System Life:  10
   Equivalent Weapon Systems Per Intermediate Location:      1
   System Operation Hours Per Month:  120
   SE Development Cost:     1000
   Sensitivity Alternatives:  4
   Lower Sensitivity Range:  0.50
   Upper Sensitivity Range:  2.00
   Sensitivity Type:  1
   Optional Information:  INPUT SCREENS TEST
```



```
Exit Display Module (Y or N)?   N
```



```
             RECORD TYPE 2
        MAINTENANCE SYSTEM DATA
Intermediate Shop Man-hours:  145
Intermediate Labor Rate:  25.67
Depot Shop Man-hours:  145
Depot Labor Rate:  35.98
Intermediate Turnover Rate:  0.244
Depot Turnover Rate:  0.060
```



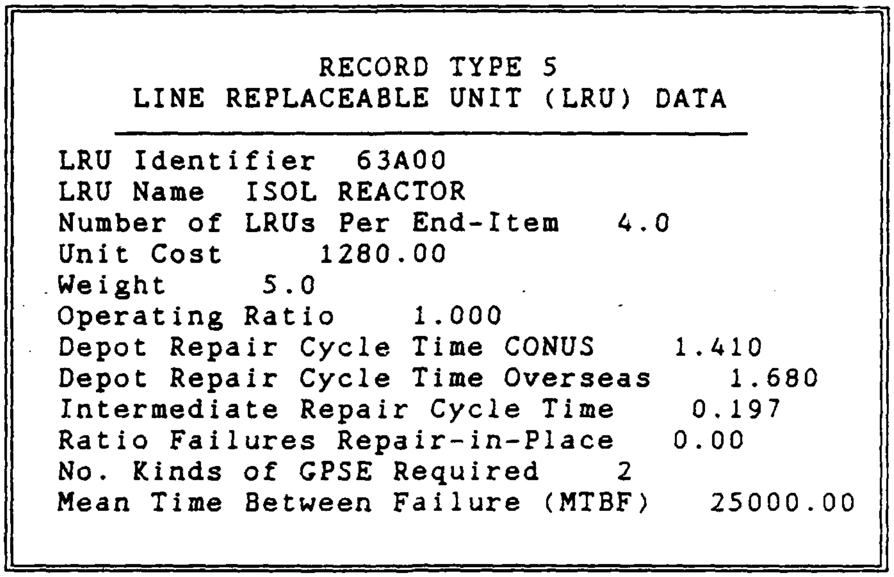
```
Exit Display Module (Y or N)?   N
```

```
           RECORD TYPE 3
        SUPPLY SYSTEM DATA

Initial Management Cost:  1200.00
Recurring Management Cost:   150.00
Base Supply Management Cost:    9.870
Order and Ship Time CONUS:  0.329
Order and Ship Time Overseas:  0.472
Packing Cost CONUS:    2.320
Packing Cost Overseas:    2.320
Packed Wt Ratio CONUS:    1.940
Packed Wt Ratio Overseas:    1.940
Shipping Rate CONUS:    0.420
Shipping Rate Overseas:    1.600
Tech Data Cost:  617.00
```

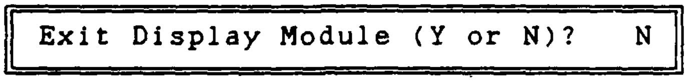
```
Exit Display Module (Y or N)?   N
```

```
           RECORD TYPE 4
      SUPPORT EQUIPMENT DATA

SE Resource Number (SERN)  8006
SE Name  SIG.GEN.SFW*
SE Cost      2500.00
SE Operating and Maintenance Cost     100.00
Number of SE   0
Current Usage    0.00
Available SE Time  200.00
Facilities Cost          0.00
```

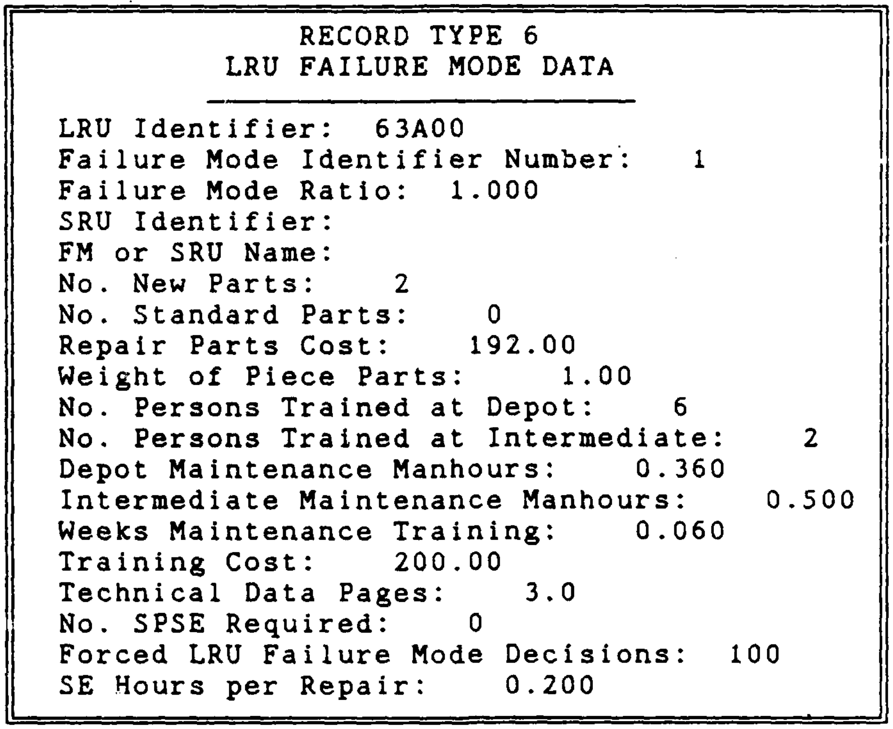
```
Exit Display Module (Y or N)?   N
```



```
                RECORD TYPE 5
     LINE REPLACEABLE UNIT (LRU) DATA

LRU Identifier  63A00
LRU Name  ISOL REACTOR
Number of LRUs Per End-Item   4.0
Unit Cost     1280.00
Weight     5.0
Operating Ratio    1.000
Depot Repair Cycle Time CONUS     1.410
Depot Repair Cycle Time Overseas     1.680
Intermediate Repair Cycle Time    0.197
Ratio Failures Repair-in-Place   0.00
No. Kinds of GPSE Required    2
Mean Time Between Failure (MTBF)   25000.00
```



```
Exit Display Module (Y or N)?   N
```



```
             RECORD TYPE 6
         LRU FAILURE MODE DATA

LRU Identifier:  63A00
Failure Mode Identifier Number:    1
Failure Mode Ratio:  1.000
SRU Identifier:
FM or SRU Name:
No. New Parts:    2
No. Standard Parts:    0
Repair Parts Cost:    192.00
Weight of Piece Parts:     1.00
No. Persons Trained at Depot:    6
No. Persons Trained at Intermediate:     2
Depot Maintenance Manhours:    0.360
Intermediate Maintenance Manhours:     0.500
Weeks Maintenance Training:    0.060
Training Cost:    200.00
Technical Data Pages:    3.0
No. SPSE Required:    0
Forced LRU Failure Mode Decisions:  100
SE Hours per Repair:    0.200
```

```
                       RECORD TYPE 7
              SHOP REPLACEABLE UNIT (SRU) DATA
           ____________________________________

 SRU Identifier:  63DA0          SRU Cost:    698.00
 Weight of SRU:    0.5
 Cost of Piece Parts and Assys Per SRU Repair:    209.00
 Weight of Parts and Assemblies:    0.01
 No. of New Piece Parts and Assemblies Used:    1
 No. Standard Piece Parts per Repair:    0
 No. Pages of Tech Data per Repair:    7.0
 No. Kinds of SE Used per SRU Repair:   2
 Depot Repair Cycle Time CONUS:    1.650
 Depot Repair Cycle Time Overseas:    1.910
 Intermediate Repair Cycle Time:    0.263
 Depot Maintenance Manhours per Repair:    0.750
 Intermediate Maintenance Manhours per Repair:    0.750
 Depot Trainees:    0
 Intermediate Trainees:    0
 Maintenance Training Weeks:    0.060
 Maintenance Training Cost:    200.00
 Forced SRU Decisions:    0
 SE Hours per Repair:    0.750
```

```
                          RECORD TYPE 8
                 LRU/SRU/SE CROSS REFERENCE DATA
              ____________________________________

 Record Type Identifier:  3
 LRU or SRU Identifier:  63A00
 Failure Mode Identification Number:
 SE Resource Numbers:  1001      SE Resource Numbers:
 SE Resource Numbers:  5001      SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 SE Resource Numbers:            SE Resource Numbers:
 Continuation Indicator:
```

```
 Exit Display Module (Y or N)?   N
```

```
          RECORD TYPE 9
       OUTPUT OPTIONS DATA
     _______________________

Echo of Input Factors:  0
Support Equipment Input Values:  0
SE Cross-Reference Table:  0
Computed SE Costs:  0
SE Requirements:  0
Repair Level Decisions Details:  0
LRU Repair Location Summary:  0
Desired Output Units:  1
```

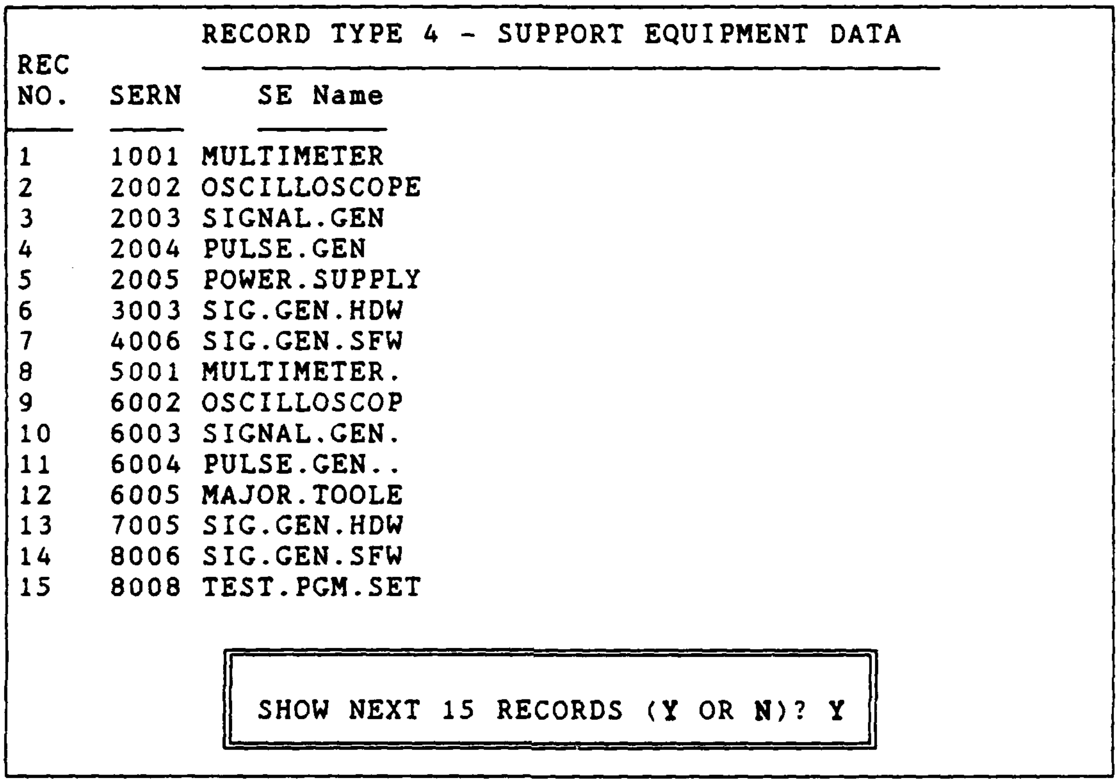
```
Exit Display Module (Y or N)?   N
```

```
                RECORD TYPE 10
           WHOLESALE CHANGE FACTORS
          __________________________

Multiply all LRU and LRU piece part costs
     and SRU and SRU piece part costs:      1.000
Multiply all LRU MTBFS:     1.000
Multiply all SE costs:     1.000
```

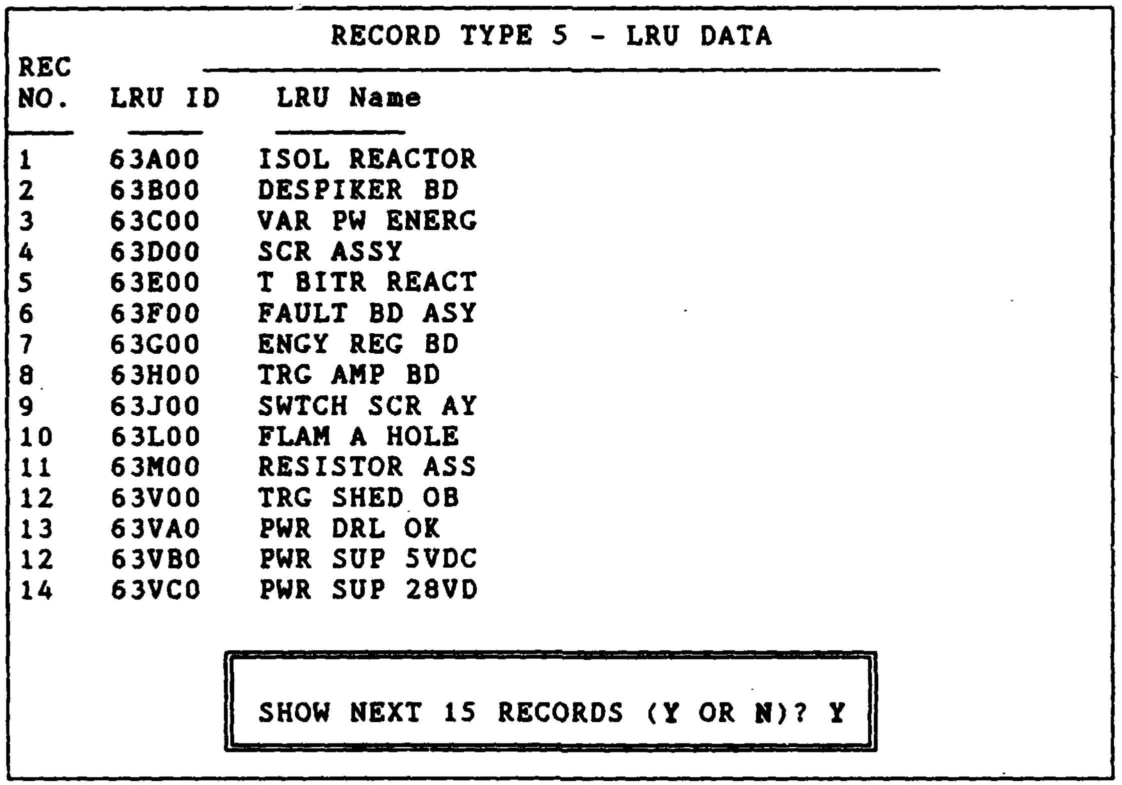
```
              EXCLUSION FACTORS
             ___________________

Exclude depot repair for all LRUs:  0
Exclude scrap for all LRUS:  0
Exclude intermediate repair for all LRUs:  0
Exclude depot repair for all SRUs:  0
Exclude scrap for all SRUs:  0
Exclude intermediate for all SRUs:  0
```

```
              RECORD TYPE 11
         INDIVIDUAL CHANGE FACTORS
        ___________________________

SE Resource Numbers:  2002
SE Desired Change Factor:     1.50
SE Resource Numbers:  6002
SE Desired Change Factor:     1.50
LRU Identifier (work unit code):  63L00
LRU Cost Factor:      1.50
LRU MTBF:      0.75
```

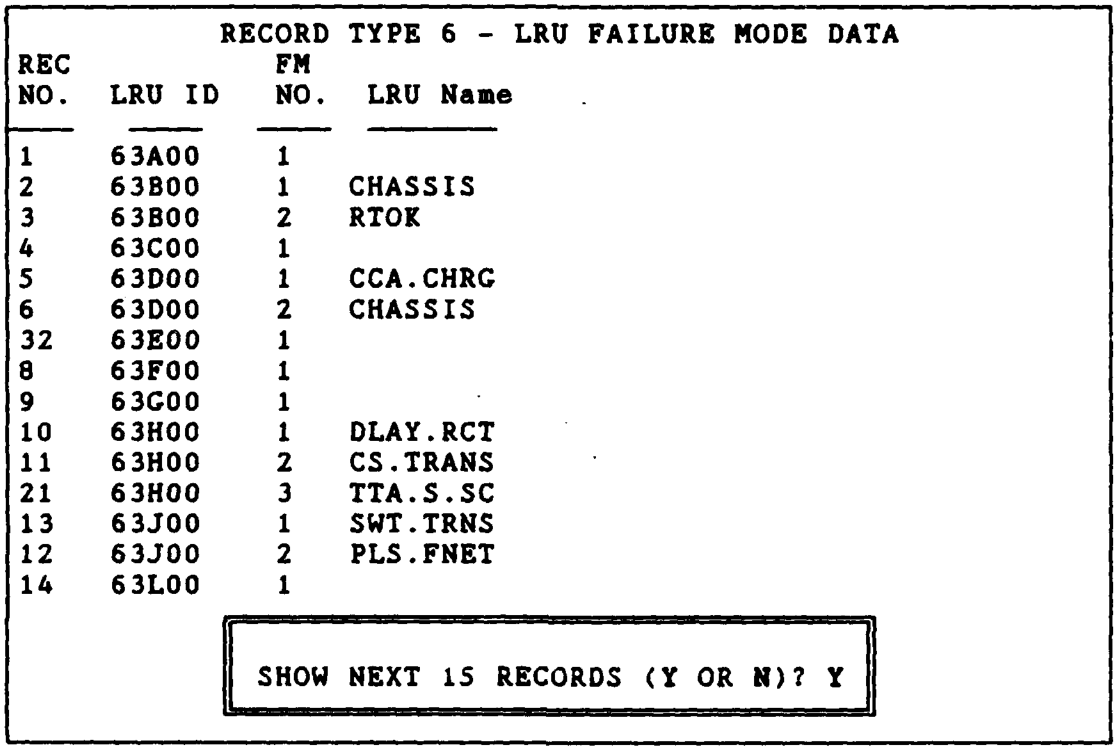
Appendix C: View Data Record Screens



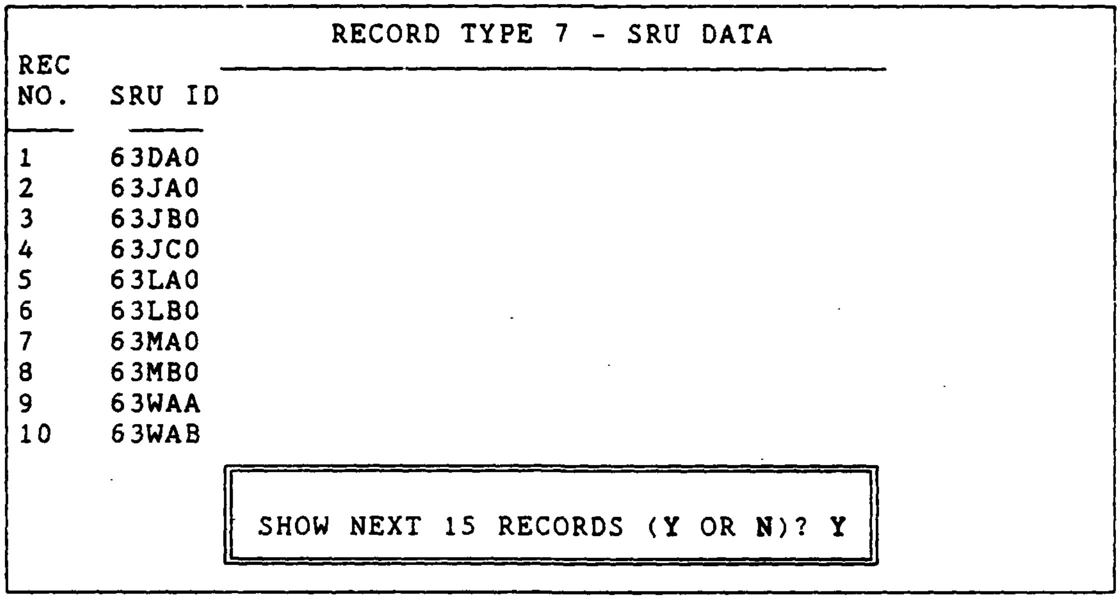
```
                RECORD TYPE 4 - SUPPORT EQUIPMENT DATA
REC
NO.     SERN       SE Name

1       1001 MULTIMETER
2       2002 OSCILLOSCOPE
3       2003 SIGNAL.GEN
4       2004 PULSE.GEN
5       2005 POWER.SUPPLY
6       3003 SIG.GEN.HDW
7       4006 SIG.GEN.SFW
8       5001 MULTIMETER.
9       6002 OSCILLOSCOP
10      6003 SIGNAL.GEN.
11      6004 PULSE.GEN..
12      6005 MAJOR.TOOLE
13      7005 SIG.GEN.HDW
14      8006 SIG.GEN.SFW
15      8008 TEST.PGM.SET

           SHOW NEXT 15 RECORDS (Y OR N)? Y
```



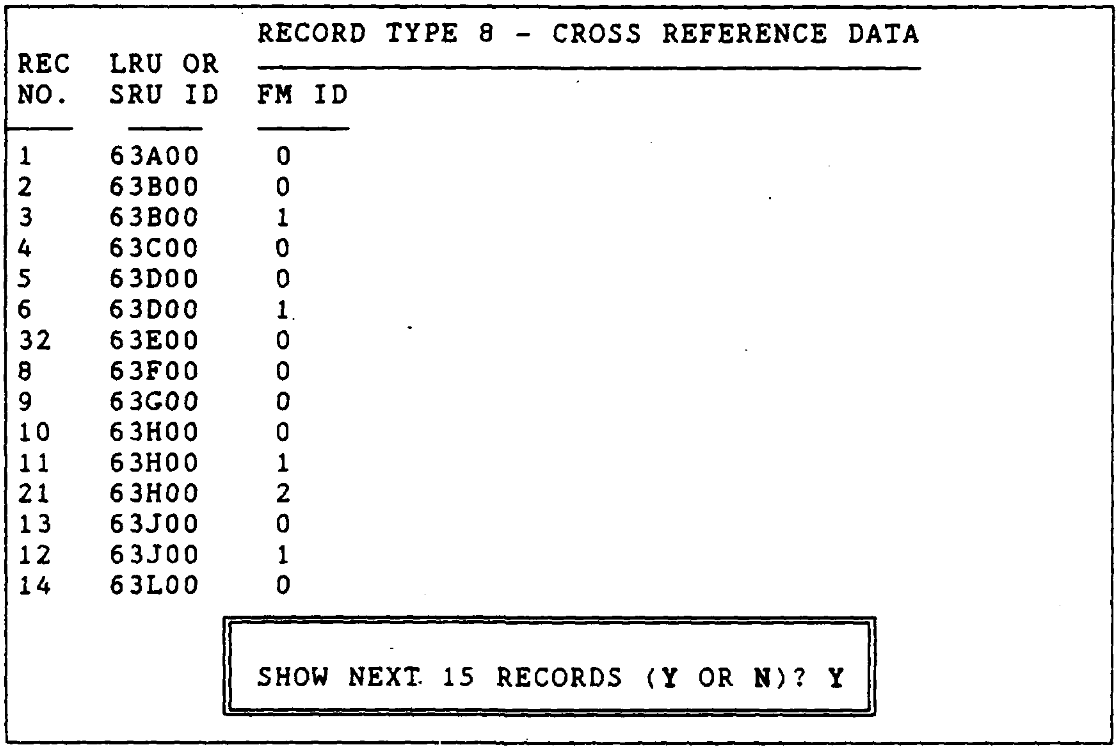
```
                   RECORD TYPE 5 - LRU DATA
REC
NO.   LRU ID   LRU Name
----  -----    --------
1     63A00    ISOL REACTOR
2     63B00    DESPIKER BD
3     63C00    VAR PW ENERG
4     63D00    SCR ASSY
5     63E00    T BITR REACT
6     63F00    FAULT BD ASY
7     63G00    ENGY REG BD
8     63H00    TRG AMP BD
9     63J00    SWTCH SCR AY
10    63L00    FLAM A HOLE
11    63M00    RESISTOR ASS
12    63V00    TRG SHED OB
13    63VA0    PWR DRL OK
12    63VB0    PWR SUP 5VDC
14    63VC0    PWR SUP 28VD

            SHOW NEXT 15 RECORDS (Y OR N)? Y
```



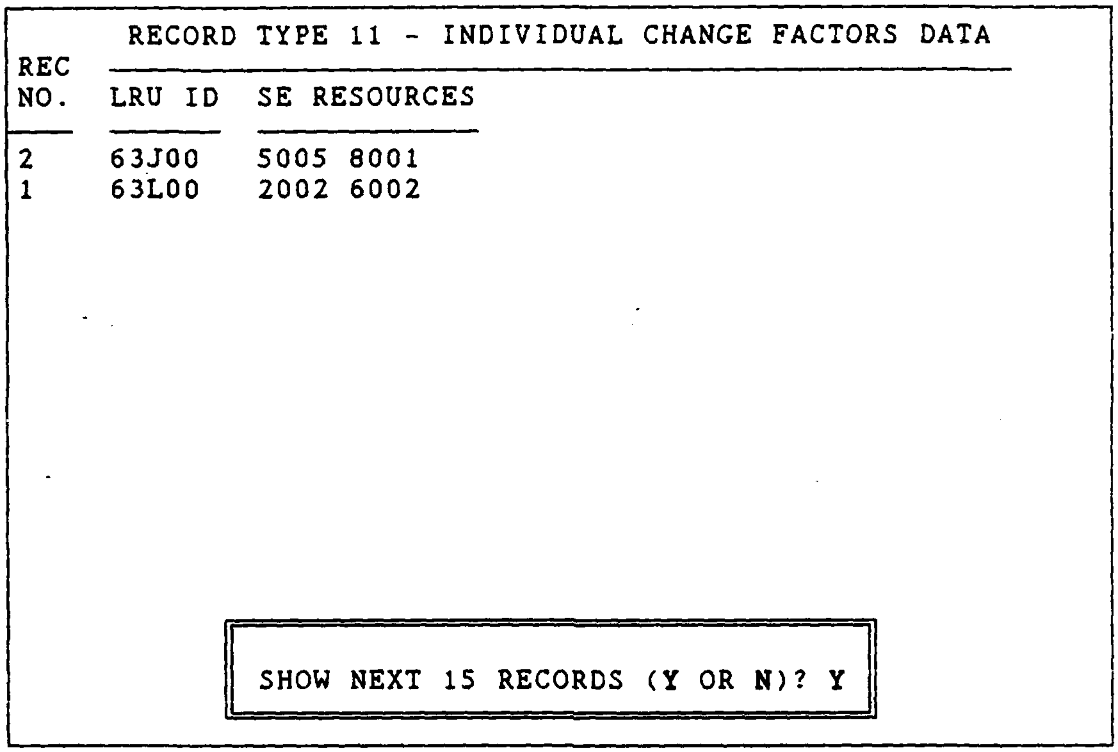
```
              RECORD TYPE 6 - LRU FAILURE MODE DATA
REC             FM
NO.   LRU ID    NO.   LRU Name
----  -----    ----   --------
1     63A00     1
2     63B00     1    CHASSIS
3     63B00     2    RTOK
4     63C00     1
5     63D00     1    CCA.CHRG
6     63D00     2    CHASSIS
32    63E00     1
8     63F00     1
9     63G00     1
10    63H00     1    DLAY.RCT
11    63H00     2    CS.TRANS
21    63H00     3    TTA.S.SC
13    63J00     1    SWT.TRNS
12    63J00     2    PLS.FNET
14    63L00     1

            SHOW NEXT 15 RECORDS (Y OR N)? Y
```



```
                    RECORD TYPE 7 - SRU DATA
REC            ______________________________________
NO.   SRU ID
___   ______

1     63DA0
2     63JA0
3     63JB0
4     63JC0
5     63LA0
6     63LB0
7     63MA0
8     63MB0
9     63WAA
10    63WAB

           SHOW NEXT 15 RECORDS (Y OR N)? Y
```



```
               RECORD TYPE 8 - CROSS REFERENCE DATA
REC   LRU OR   ______________________________________
NO.   SRU ID   FM ID
___   ______   _____

1     63A00     0
2     63B00     0
3     63B00     1
4     63C00     0
5     63D00     0
6     63D00     1
32    63E00     0
8     63F00     0
9     63G00     0
10    63H00     0
11    63H00     1
21    63H00     2
13    63J00     0
12    63J00     1
14    63L00     0

           SHOW NEXT 15 RECORDS (Y OR N)? Y
```



```
         RECORD TYPE 11 - INDIVIDUAL CHANGE FACTORS DATA
REC      ________________________________________________
NO.      LRU ID    SE RESOURCES
___      ______    ____________
2        63J00     5005 8001
1        63L00     2002 6002




            SHOW NEXT 15 RECORDS (Y OR N)? Y
```

Appendix D: Program Coding

```
*NRLAFACE.PRG
SET COLOR TO B/W
SET BELL OFF
SET CONFIRM ON
clear
set talk off
STORE SPACE(1) TO CHOICE
DO COPYRGHT
DO WHILE .T.
SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@2,20 TO 17,58 DOUBLE
@19,21 TO 21,57 DOUBLE
@6,24 TO 6,54

SET COLOR TO N/W
@4,25 say 'NETWORK REPAIR LEVEL ANALYSIS'
@ROW()+1,35 SAY 'MAIN MENU'
SET COLOR TO B/W
@ROW()+2,28 SAY 'ISPLAY CURRENT DATA FILE'
@ROW()+2,30 SAY 'DIT CURRENT DATA FILE'
@ROW()+2,31 SAY 'REATE NEW DATA FILE'
@ROW()+2,31 SAY 'UN ANALYSIS MODELS'
@ROW()+2,24 SAY 'E'
@ROW(),26 SAY 'IT MENU (TO OPERATING SYSTEM)'
@ROW()+5,24 SAY 'ENTER '
@ROW(),42 SAY ' SELECTION: '
@ROW(),54 GET CHOICE PICTURE '@!'

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/W
ENDIF
@7,27 SAY 'D'
@ROW()+2,29 SAY 'E'
@ROW()+2,30 SAY 'C'
@ROW()+2,30 SAY 'R'
@ROW()+2,25 SAY 'X'
@ROW()+5,30 SAY 'HIGH-LIGHTED'

READ
set color to w/n
do case
  case choice = 'D'
   do display
  case choice = 'E'
   do edit
  case choice = 'C'
```

```
   do create
  case choice = 'R'
   do run
  case choice = 'X'
   clear
   quit
  case choice = '6'
   clear
clear
set talk on
SET CONFIRM OFF
cancel
endcase
ENDDO
set color to w/n
^Z

**********************************************************
**********************************************************
*COPYRGHT.PRG -

SET BELL OFF
set talk off
STORE SPACE(1) TO GOON

SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@4,6 TO 20,74 DOUBLE

SET COLOR TO R/W
@2,32 SAY 'NRLA INTERFACE'
SET COLOR TO N/W
@ROW()+1,29 SAY 'RIGHTS AND WARRANTEES'
@ROW()+2,27 SAY 'RESTRICTED RIGHTS WARNING'
SET COLOR TO G/W
@ROW()+1,26 SAY '---------------------------'
SET COLOR TO B/W
@ROW()+1,12 SAY 'NRLA INTERFACE is a copyrighted package
designed for the'
@ROW()+1,14 SAY 'exclusive use of the United States Military,
and is '
@ROW()+1,9 SAY 'protected by U.S. Copyright Law (Title 17
United States Code).'
@ROW()+1,8 SAY 'Unauthorized reproduction and/or sales may
result in imprisonment'
@ROW()+1,11 SAY 'of up to ONE YEAR, and fines of up to
$10,000 (17 USC 506).'
@ROW()+1,11 SAY 'Copyright infringers may also be subject to
civil liability.'
SET COLOR TO N/W
@ROW()+2,28 SAY 'DISCLAIMER OF WARRANTEE'
```

```
SET COLOR TO G/W
@ROW()+1,26 SAY '----------------------------'
SET COLOR TO B/W
@ROW()+1,8 SAY 'This software and manual is distributed
without any expressed or'
@ROW()+1,11 SAY 'implied warrantees whatsoever.  The user is
advised to test'
@ROW()+1,18 SAY 'the program thoroughly before relying on
it.'
@ROW()+1,13 SAY 'The user assumes the entire risk of using
this program.'
SET COLOR TO N/W
@ROW()+2,27 SAY 'Press any key to continue.'
WAIT""
RETURN
^Z

**********************************************************
**********************************************************
*DISPLAY.PRG
SET COLOR TO B/W
SET BELL OFF
clear
set talk off
STORE SPACE(1) TO NUMB

DO WHILE .T.
SET CONFIRM ON
SET COLOR TO G/W
@1,14 TO 19,65 DOUBLE
@4,17 to 4,62
@21,21 TO 23,59 DOUBLE

SET COLOR TO N/W
@3,18 say 'WHICH RECORD TYPE WOULD YOU LIKE TO DISPLAY?'

SET COLOR TO B/W
@5,31 SAY 'EAPONS SYSTEM DATA'
@ROW()+1,29 SAY 'AINTENANCE SYSTEM DATA'
@ROW()+1,31 SAY 'S'
@ROW(),33 SAY 'PPLY SYSTEM DATA'
@ROW()+1,29 SAY 'SUPPORT '
@ROW(),38 SAY 'QUIPMENT DATA'
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,29 SAY 'LRU '
@ROW(),34 SAY 'AILURE MODE DATA'
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,24 SAY 'LRU/SRU/SE '
@ROW(),36 SAY 'ROSS REFERENCE DATA'
@ROW()+1,34 SAY 'UTPUT OPTIONS'
@ROW()+1,31 SAY 'W'
@ROW(),33 SAY 'OLESALE FACTORS'
```

```
@ROW()+1,32 SAY 'NDIVIDUAL FACTORS'
@ROW()+1,33 SAY 'LL DATA RECORDS'
@ROW()+1,31 SAY 'ETURN TO MAIN MENU'
@22,25 SAY 'ENTER '
@22,43 SAY ' SELECTION: '
@22,55 GET NUMB PICTURE '@!'

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/W
ENDIF
@5,30 SAY 'W'
@ROW()+1,28 SAY 'M'
@ROW()+1,32 SAY 'U'
@ROW()+1,37 SAY 'E'
@ROW()+1,36 SAY 'L'
@ROW()+1,33 SAY 'F'
@ROW()+1,36 SAY 'S'
@ROW()+1,35 SAY 'C'
@ROW()+1,33 SAY 'O'
@ROW()+1,32 SAY 'H'
@ROW()+1,31 SAY 'I'
@ROW()+1,32 SAY 'A'
@ROW()+1,30 SAY 'R'
@22,31 SAY 'HIGH-LIGHTED'


READ
do case
 case numb = 'W'
  do dweapsys
 case numb = 'M'
  do dmaintsys
 case numb = 'U'
  do dsupsys
 case numb = 'E'
  do dsesys
 case numb = 'L'
  do dlru
 CASE NUMB = 'F'
  DO DLRUFM
 CASE NUMB = 'S'
  DO DSRU
 CASE NUMB = 'C'
  DO DCROSS
 CASE NUMB = 'O'
  DO DOUTOPT
 CASE NUMB = 'H'
  DO DWHOLFAC
 CASE NUMB = 'I'
  DO DINDFAC
 CASE NUMB = 'A'
```

```
  DO DALL
 CASE NUMB = 'R'
  RETURN TO MASTER
 ENDCASE
ENDDO
SET CONFIRM OFF
^Z

***********************************************************
***********************************************************

*EXIT.PRG -
*EXITS PROGRAM CONTROL FROM DISPLAY MODULE
SET CONSOLE ON
     STORE 'N' TO EXIT
     SET COLOR TO G/W
     @22,23 TO 24,59 DOUBLE
     @23,24 CLEAR TO 23,60
     SET COLOR TO B/W
     @23,25 SAY 'Exit Display Module (Y or N)?'
     @23,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          CLOSE DATABASES
          CLEAR
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          CLEAR
          RETURN
       ENDCASE
^Z

***********************************************************
***********************************************************

*EMPTY.PRG -  NOTIFIES USER OF AN EMPTY DATABASE FILE

SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@8,14 TO 12,65 DOUBLE

SET COLOR TO B/W
@10,21 say 'NO DATA IS STORED IN THIS RECORD TYPE'
?
?
?
?

WAIT
^Z************************************************************
```

```
***********************************************************
*DWEAPSYS.PRG - DISPLAYS WEAPONS SYSTEM DATA
SET COLOR TO B/W
CLEAR
USE WEAPON_SY

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@1,8 TO 19,70 DOUBLE
@5,30 TO 5,49
SET COLOR TO N/W
@3,33 SAY 'RECORD TYPE 1'
@ROW()+1,31 SAY 'WEAPON SYSTEM DATA'
SET COLOR TO B/W
@ROW()+2,10 SAY ' End Item Name: ' GET END_ITEM
@ROW()+1,10 SAY ' No. of Intermediate Locations: ' GET
NO_I_LOC
@ROW()+1,10 SAY ' Ratio Force Overseas: ' GET RATIO_OS
@ROW()+1,10 SAY ' Years System Life: ' GET YRS_SYS_LI
@ROW()+1,10 SAY ' Equivalent Weapon Systems Per Intermediate
Location: ' GET EQ_WS_ILOC
@ROW()+1,10 SAY ' System Operation Hours Per Month: ' GET
MO_OPS_HRS
@ROW()+1,10 SAY ' SE Development Cost: ' GET SE_DEV_CST

@ROW()+1,10 SAY ' Sensitivity Alternatives: ' GET SENSI_ALT

@ROW()+1,10 SAY ' Lower Sensitivity Range: ' GET L_SEN_RANGE

@ROW()+1,10 SAY ' Upper Sensitivity Range: ' GET U_SEN_RANGE

@ROW()+1,10 SAY ' Sensitivity Type: ' GET SENSI_TYPE
@ROW()+1,10 SAY ' Optional Information: ' GET OPTION_INF

CLEAR GETS
?
?
?
SKIP
ENDDO
DO EXIT
CLOSE DATABASES
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*DMAINTSY.PRG - DISPLAYS MAINTENANCE SYSTEM DATA
SET COLOR TO B/W
CLEAR
USE MAINT_SYS

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@5,22 TO 17,61 DOUBLE
@9,29 TO 9,53
SET COLOR TO N/W
@7,35 SAY 'RECORD TYPE 2'
@ROW()+1,30 SAY 'MAINTENANCE SYSTEM DATA'
SET COLOR TO B/W
@ROW()+2,24 SAY ' Intermediate Shop Man-hours: ' GET
INT_MNHRS
@ROW()+1,24 SAY ' Intermediate Labor Rate: ' GET INT_LAB_RT
@ROW()+1,24 SAY ' Depot Shop Man-hours: ' GET DEP_MNHRS
@ROW()+1,24 SAY ' Depot Labor Rate: ' GET DEP_LAB_RT
@ROW()+1,24 SAY ' Intermediate Turnover Rate: ' GET
INT_TRN_RT
@ROW()+1,24 SAY ' Depot Turnover Rate: ' GET DEP_TRN_RT
CLEAR GETS
?
?
?
?
?
SKIP
ENDDO
DO EXIT
CLOSE DATABASES
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*DSUPSYS.PRG - DISPLAYS SUPPLY SYSTEM DATA
SET COLOR TO B/W
CLEAR
USE SUPPLY_SYS

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@2,21 TO 20,62 DOUBLE
@6,31 TO 6,50
SET COLOR TO N/W
@4,35 SAY 'RECORD TYPE 3'
@ROW()+1,32 SAY 'SUPPLY SYSTEM DATA'
SET COLOR TO B/W
@ROW()+2,23 SAY ' Initial Management Cost: ' GET INI_MGT_CS
@ROW()+1,23 SAY ' Recurring Management Cost: ' GET REC_MGT_CS
@ROW()+1,23 SAY ' Base Supply Management Cost: ' GET
BS_MGT_CS
@ROW()+1,23 SAY ' Order and Ship Time CONUS: ' GET CONUS_SHIP
@ROW()+1,23 SAY ' Order and Ship Time Overseas: ' GET OS_SHIP
@ROW()+1,23 SAY ' Packing Cost CONUS: ' GET CONUS_PK_C
@ROW()+1,23 SAY ' Packing Cost Overseas: ' GET OS_PK_CST
@ROW()+1,23 SAY ' Packed Wt Ratio CONUS: ' GET CONUS_PKWT
@ROW()+1,23 SAY ' Packed Wt Ratio Overseas: ' GET OS_PKWT_RT
@ROW()+1,23 SAY ' Shipping Rate CONUS: ' GET CONUS_SHRT
@ROW()+1,23 SAY ' Shipping Rate Overseas: ' GET OS_SHIP_RT
@ROW()+1,23 SAY ' Tech Data Cost: ' GET TEC_DAT_CS
CLEAR GETS
?
?
?
SKIP
ENDDO
DO EXIT
CLOSE DATABASES
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*DSESYS.PRG - DISPLAYS SUPPORT EQUIPMENT DATA
SET COLOR TO B/W
CLEAR
USE SE_DATA
SET TALK OFF
IF EOF()
  DO EMPTY
ENDIF
DO WHILE .NOT. EOF()
SET COLOR TO G/W
@5,18 TO 19,64 DOUBLE
@9,28 TO 9,51
SET COLOR TO N/W
@7,34 SAY 'RECORD TYPE 4'
@ROW()+1,29 SAY 'SUPPORT EQUIPMENT DATA'
SET COLOR TO B/W
@ROW()+2,20 SAY ' SE Resource Number (SERN) ' GET SERN
@ROW()+1,20 SAY ' SE Name ' GET SE_NAME
@ROW()+1,20 SAY ' SE Cost ' GET SE_COST
@ROW()+1,20 SAY ' SE Operating and Maintenance Cost ' GET
SE_O_M_CST
@ROW()+1,20 SAY ' Number of SE ' GET NO_OF_SE
@ROW()+1,20 SAY ' Current Usage ' GET CURR_USAGE
@ROW()+1,20 SAY ' Available SE Time ' GET AVAIL_TIME
@ROW()+1,20 SAY ' Facilities Cost ' GET FACIL_COST
CLEAR GETS

SKIP
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*DLRU.PRG - DISPLAYS LINE REPLACEABLE UNIT (LRU) DATA
SET COLOR TO B/W
CLEAR
USE LRU_DATA
SET TALK OFF

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@3,18 TO 21,65 DOUBLE
@7,24 TO 7,57
SET COLOR TO N/W
@5,35 SAY 'RECORD TYPE 5'
@6,25 SAY 'LINE REPLACEABLE UNIT (LRU) DATA'
SET COLOR TO B/W
@ROW()+2,20 SAY ' LRU Identifier ' GET LRU_ID
@ROW()+1,20 SAY ' LRU Name ' GET LRU_NAME
@ROW()+1,20 SAY ' Number of LRUs Per End-Item ' GET
NO_LRU_END
@ROW()+1,20 SAY ' Unit Cost ' GET UNIT_COST
@ROW()+1,20 SAY ' Weight ' GET WEIGHT
@ROW()+1,20 SAY ' Operating Ratio ' GET OPS_RATIO
@ROW()+1,20 SAY ' Depot Repair Cycle Time CONUS ' GET
DEP_RCT_CO
@ROW()+1,20 SAY ' Depot Repair Cycle Time Overseas ' GET
DEP_RCT_OS
@ROW()+1,20 SAY ' Intermediate Repair Cycle Time ' GET
INT_RCT_CO
@ROW()+1,20 SAY ' Ratio Failures Repair-in-Place ' GET
R_FAIL_RIP
@ROW()+1,20 SAY ' No. Kinds of GPSE Required ' GET NO_GPSE_RQ
@ROW()+1,20 SAY ' Mean Time Between Failure (MTBF) ' GET MTBF
CLEAR GETS
SKIP
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************
```

```
*DLRUFM.PRG - DISPLAYS LRU FAILURE MODE DATA
SET COLOR TO B/W
CLEAR
USE LRU_FM_D
SET TALK OFF

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@0,18 TO 23,65 DOUBLE
@3,29 TO 3,51
SET COLOR TO N/W
@1,34 SAY 'RECORD TYPE 6'
@2,30 SAY 'LRU FAILURE MODE DATA'
SET COLOR TO B/W
@ROW()+2,20 SAY ' LRU Identifier: ' GET LRU_ID
@ROW()+1,20 SAY ' Failure Mode Identifier Number: ' GET
FM_IDEN_NO
@ROW()+1,20 SAY ' Failure Mode Ratio: ' GET  FM_RATIO
@ROW()+1,20 SAY ' SRU Identifier: ' GET SRU_ID
@ROW()+1,20 SAY ' FM or SRU Name: ' GET FM_SRU_NAM
@ROW()+1,20 SAY ' No. New Parts: ' GET NO_NEW_PRT
@ROW()+1,20 SAY ' No. Standard Parts: ' GET NO_STD_PRT
@ROW()+1,20 SAY ' Repair Parts Cost: ' GET REP_PRT_CS
@ROW()+1,20 SAY ' Weight of Piece Parts: ' GET WT_PC_PART
@ROW()+1,20 SAY ' No. Persons Trained at Depot: ' GET
NO_TRN_DEP
@ROW()+1,20 SAY ' No. Persons Trained at Intermediate: ' GET
NO_TRN_INT
@ROW()+1,20 SAY ' Depot Maintenance Manhours: ' GET
DEP_MX_MHR
@ROW()+1,20 SAY ' Intermediate Maintenance Manhours: ' GET
INT_MX_MHR
@ROW()+1,20 SAY ' Weeks Maintenance Training: ' GET
WKS_MX_TRN
@ROW()+1,20 SAY ' Training Cost: ' GET TRNG_CST
@ROW()+1,20 SAY ' Technical Data Pages: ' GET TEC_DT_PGS
@ROW()+1,20 SAY ' No. SPSE Required: ' GET NO_SPSE_RQ
@ROW()+1,20 SAY ' Forced LRU Failure Mode Decisions: ' GET
FRC_LRU_FM
@ROW()+1,20 SAY ' SE Hours per Repair: ' GET SE_HRS_REP
CLEAR GETS
SKIP
?
WAIT
CLEAR
DO EXIT
ENDDO
CLOSE DATABASES
```

```
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*DSRU.PRG - DISPLAYS SHOP REPLACEABLE UNIT (SRU) DATA
SET COLOR TO B/W
CLEAR
USE SRU_DATA
SET TALK OFF

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@0,13 TO 23,71 DOUBLE
@3,23 TO 3,58
SET COLOR TO N/W
@1,35 SAY 'RECORD TYPE 7'
@2,25 SAY 'SHOP REPLACEABLE UNIT (SRU) DATA'
SET COLOR TO B/W
@ROW()+2,15 SAY ' SRU Identifier: ' GET SRU_ID
@ROW(),46 SAY ' SRU Cost: ' GET SRU_CST
@ROW()+1,15 SAY ' Weight of SRU: ' GET SRU_WEIGHT
@ROW()+1,15 SAY ' Cost of Piece Parts and Assys Per SRU
Repair: ' GET CST_PP_REP
@ROW()+1,16 SAY 'Weight of Parts and Assemblies: ' GET
WT_PP_ASSY
@ROW()+1,15 SAY ' No. of New Piece Parts and Assemblies Used:
' GET NO_NEW_PP
@ROW()+1,15 SAY ' No. Standard Piece Parts per Repair: ' GET
NO_STD_PP
@ROW()+1,15 SAY ' No. Pages of Tech Data per Repair: ' GET
NO_PGS_T_D
@ROW()+1,15 SAY ' No. Kinds of SE Used per SRU Repair: ' GET
NO_KIND_SE
@ROW()+1,15 SAY ' Depot Repair Cycle Time CONUS: ' GET
DEP_RCT_CO
@ROW()+1,15 SAY ' Depot Repair Cycle Time Overseas: ' GET
DEP_RCT_OS
@ROW()+1,15 SAY ' Intermediate Repair Cycle Time: ' GET
INT_RCT
@ROW()+1,15 SAY ' Depot Maintenance Manhours per Repair: '
GET DEP_MX_MHR
@ROW()+1,15 SAY ' Intermediate Maintenance Manhours per
Repair: ' GET INT_MX_MHR
@ROW()+1,15 SAY ' Depot Trainees: ' GET DEP_TRNEES
@ROW()+1,15 SAY ' Intermediate Trainees: ' GET INT_TRNEES
```

```
@ROW()+1,15 SAY ' Maintenance Training Weeks: ' GET
MX_TRN_WKS
@ROW()+1,15 SAY ' Maintenance Training Cost: ' GET MX_TRN_CST

@ROW()+1,15 SAY ' Forced SRU Decisions: ' GET FRC_SRU_DE
@ROW()+1,15 SAY ' SE Hours per Repair: ' GET SE_HRS_REP
CLEAR GETS
SKIP
?
WAIT
CLEAR
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*DCROSS.PRG - DISPLAYS LRU/SRU/SE CROSS REFERENCE DATA
SET COLOR TO B/W
CLEAR
USE CROSS_RE
SET TALK OFF

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@2,10 TO 20,72 DOUBLE
@6,25 to 6,57
SET COLOR TO N/W
@4,35 SAY 'RECORD TYPE 8'
@5,26 SAY 'LRU/SRU/SE CROSS REFERENCE DATA'
SET COLOR TO B/W
@ROW()+2,12 SAY ' Record Type Identifier: ' GET REC_TYP_ID
@ROW()+1,12 SAY ' LRU or SRU Identifier: ' GET LRU_SRU_ID
@ROW()+1,12 SAY ' Failure Mode Identification Number: ' GET
FM_ID_NO
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO1
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO2
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO3
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO4
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO5
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO6
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO7
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO8
@ROW()-7,43 SAY ' SE Resource Numbers: ' GET SE_RES_NO9
```

```
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N10
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N11
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N12
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N13
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N14
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N15
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N16
@ROW()+1,12 SAY ' Continuation Indicator: ' GET CONTIN_IND
CLEAR GETS
SKIP
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z


**********************************************************
**********************************************************

*DOUTOPT.PRG - DISPLAYS OUTPUT OPTIONS DATA
SET COLOR TO B/W
CLEAR
USE OUTP_OPT

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@5,22 TO 19,61 DOUBLE
@9,31 TO 9,51
SET COLOR TO N/W
@7,35 SAY 'RECORD TYPE 9'
@8,32 SAY 'OUTPUT OPTIONS DATA'
SET COLOR TO B/W
@ROW()+2,24 SAY ' Echo of Input Factors: ' GET INPUT_ECHO
@ROW()+1,24 SAY ' Support Equipment Input Values: ' GET
SE_INPUT
@ROW()+1,24 SAY ' SE Cross-Reference Table: ' GET SE_CRS_REF

@ROW()+1,24 SAY ' Computed SE Costs: ' GET SE_CSTS
@ROW()+1,24 SAY ' SE Requirements: ' GET SE_RQMTS
@ROW()+1,24 SAY ' Repair Level Decisions Details: ' GET
RL_DEC_DET
@ROW()+1,24 SAY ' LRU Repair Location Summary: ' GET
LRU_RL_SUM
@ROW()+1,24 SAY ' Desired Output Units: ' GET OUTP_UNITS
CLEAR GETS
SKIP
```

```
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z

***********************************************************
***********************************************************

*DWHOLFAC.PRG - DISPLAYS WHOLESALE EXCLUSION FACTORS DATA
SET COLOR TO B/W
CLEAR
USE WC_EX_FC

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@1,16 TO 11,69 DOUBLE
@12,16 TO 23,69 DOUBLE
@5,28 TO 5,53
@15,31 TO 15,49
SET COLOR TO N/W
@3,34 SAY 'RECORD TYPE 10'
@4,29 SAY 'WHOLESALE CHANGE FACTORS'
SET COLOR TO B/W
@ROW()+2,18 SAY ' Multiply all LRU and LRU piece part costs'

@ROW()+1,23 SAY ' and SRU and SRU piece part costs: ' GET
LRU_SRU_CS
@ROW()+1,18 SAY ' Multiply all LRU MTBFS: ' GET LRU_MTBFS
@ROW()+1,18 SAY ' Multiply all SE costs: ' GET SE_CSTS
SET COLOR TO N/W
@14,32 SAY 'EXCLUSION FACTORS'
SET COLOR TO B/W
@ROW()+2,18 SAY ' Exclude depot repair for all LRUs: ' GET
X_LRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all LRUS: ' GET
X_LRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate repair for all LRUs: '
GET X_LRU_IREP
@ROW()+1,18 SAY ' Exclude depot repair for all SRUs: ' GET
X_SRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all SRUs: ' GET
X_SRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate for all SRUs: ' GET
X_SRU_IREP
CLEAR GETS
SKIP
```

```
?
?
WAIT
CLEAR
DO EXIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*DINDFAC.PRG - DISPLAYS INDIVIDUAL CHANGE FACTORS
SET COLOR TO B/W
CLEAR
USE IND_CNGE
SET TALK OFF

IF EOF()
  DO EMPTY
ENDIF

DO WHILE .NOT. EOF()
SET COLOR TO G/W
@5,18 TO 18,63 DOUBLE
@9,27 TO 9,53
SET COLOR TO N/W
@7,34 SAY 'RECORD TYPE 11'
@8,28 SAY 'INDIVIDUAL CHANGE FACTORS'
SET COLOR TO B/W
@ROW()+2,20 SAY ' SE Resource Numbers: ' GET SE_RES_1
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET SE_CH_FAC1
@ROW()+1,20 SAY ' SE Resource Numbers: ' GET SE_RES_2
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET SE_CH_FAC2
@ROW()+1,20 SAY ' LRU Identifier (work unit code): ' GET
LRU_ID
@ROW()+1,20 SAY ' LRU Cost Factor: ' GET LRU_CST_FA
@ROW()+1,20 SAY ' LRU MTBF: ' GET MTBF_FAC
CLEAR GETS
SKIP
?
?
?
WAIT
ENDDO
CLOSE DATABASES
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*DALL.PRG - DISPLAYS ALL DATA
CLEAR
DO DWEAPSYS
DO DMAINTSYS
DO DSUPSYS
DO DSESYS
DO DLRU
DO DLRUFM
DO DSRU
DO DCROSS
DO DOUTOPT
DO DWHOLFAC
DO DINDFAC
CLOSE ALL
RETURN TO MASTER
^Z

**********************************************************
**********************************************************

*EDIT.PRG - EDIT MENU - 27-4-88

SET BELL OFF
set talk off
STORE SPACE(1) TO NUMB

DO WHILE .T.
SET CONFIRM ON
SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@1,14 TO 18,65 DOUBLE
@4,17 to 4,62
@21,21 TO 23,59 DOUBLE

SET COLOR TO N/W
@3,20 say 'WHICH RECORD TYPE WOULD YOU LIKE TO EDIT?'
SET COLOR TO B/W
@5,31 SAY 'EAPONS SYSTEM DATA'
@ROW()+1,29 SAY 'AINTENANCE SYSTEM DATA'
@ROW()+1,31 SAY 'S'
@ROW(),33 SAY 'PPLY SYSTEM DATA'
@ROW()+1,29 SAY 'SUPPORT '
@ROW(),38 SAY 'QUIPMENT DATA'
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,29 SAY 'LRU '
@ROW(),34 SAY 'AILURE MODE DATA'
```

```
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,24 SAY 'LRU/SRU/SE '
@ROW(),36 SAY 'ROSS REFERENCE DATA'
@ROW()+1,34 SAY 'UTPUT OPTIONS'
@ROW()+1,31 SAY 'W'
@ROW(),33 SAY 'OLESALE FACTORS'
@ROW()+1,32 SAY 'NDIVIDUAL FACTORS'
@ROW()+1,31 SAY 'ETURN TO MAIN MENU'
@22,25 SAY 'ENTER '
@22,43 SAY ' SELECTION: '
@22,55 GET NUMB PICTURE '@!'

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/W
ENDIF
@5,30 SAY 'W'
@ROW()+1,28 SAY 'M'
@ROW()+1,32 SAY 'U'
@ROW()+1,37 SAY 'E'
@ROW()+1,36 SAY 'L'
@ROW()+1,33 SAY 'F'
@ROW()+1,36 SAY 'S'
@ROW()+1,35 SAY 'C'
@ROW()+1,33 SAY 'O'
@ROW()+1,32 SAY 'H'
@ROW()+1,31 SAY 'I'
@ROW()+1,30 SAY 'R'
@22,31 SAY 'HIGH-LIGHTED'


READ
do case
 case numb = 'W'
  do eweapsys
 case numb = 'M'
  do emaintsys
 case numb = 'U'
  do esupsys
 case numb = 'E'
  do esesys
 case numb = 'L'
  do elru
 CASE NUMB = 'F'
  DO ELRUFM
 CASE NUMB = 'S'
  DO ESRU
 CASE NUMB = 'C'
  DO ECROSS
 CASE NUMB = 'O'
  DO EOUTOPT
 CASE NUMB = 'H'
```

```
  DO EWHOLFAC
 CASE NUMB = 'I'
  DO EINDFAC
 CASE NUMB = 'R'
  RETURN TO MASTER
 ENDCASE
ENDDO
SET CONFIRM ON
^Z

**********************************************************
**********************************************************

*EWEAPSYS.PRG - EDITS WEAPONS SYSTEM DATA - 27-4-88
SET COLOR TO W/B
CLEAR
USE WEAPON_SY
SET CONFIRM ON
SET COLOR TO G/B
@3,8 TO 21,70 DOUBLE
@7,30 TO 7,49
SET COLOR TO W/B
@5,33 SAY 'RECORD TYPE 1'
@ROW()+1,31 SAY 'WEAPON SYSTEM DATA'
SET COLOR TO W/B
IF EOF()
 APPEND BLANK
ENDIF

@ROW()+2,10 SAY ' End Item Name: ' GET END_ITEM
@ROW()+1,10 SAY ' No. of Intermediate Locations: ' GET
NO_I_LOC
@ROW()+1,10 SAY ' Ratio Force Overseas: ' GET RATIO_OS
@ROW()+1,10 SAY ' Years System Life: ' GET YRS_SYS_LI
@ROW()+1,10 SAY ' Equivalent Weapon Systems Per Intermediate
Location: ' GET EQ_WS_ILOC
@ROW()+1,10 SAY ' System Operation Hours Per Month: ' GET
MO_OPS_HRS
@ROW()+1,10 SAY ' SE Development Cost: ' GET SE_DEV_CST
@ROW()+1,10 SAY ' Sensitivity Alternatives: ' GET SENSI_ALT
@ROW()+1,10 SAY ' Lower Sensitivity Range: ' GET L_SEN_RANGE
@ROW()+1,10 SAY ' Upper Sensitivity Range: ' GET U_SEN_RANGE
@ROW()+1,10 SAY ' Sensitivity Type: ' GET SENSI_TYPE
@ROW()+1,10 SAY ' Optional Information: ' GET OPTION_INF
READ
?
?
?
*WAIT
SET CONFIRM OFF
CLOSE DATABASES
SET COLOR TO B/W
```

```
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*EMAINTSY.PRG - EDITS MAINTENANCE SYSTEM DATA 27-4-88
SET COLOR TO W/B
CLEAR
USE MAINT_SYS
SET CONFIRM ON
IF EOF()
 APPEND BLANK
ENDIF
SET COLOR TO G/B
@7,22 TO 19,61 DOUBLE
@11,29 TO 11,53
SET COLOR TO W/B
@9,35 SAY 'RECORD TYPE 2'
@ROW()+1,30 SAY 'MAINTENANCE SYSTEM DATA'
@ROW()+2,24 SAY ' Intermediate Shop Man-hours: ' GET
INT_MNHRS
@ROW()+1,24 SAY ' Intermediate Labor Rate: ' GET INT_LAB_RT
@ROW()+1,24 SAY ' Depot Shop Man-hours: ' GET DEP_MNHRS
@ROW()+1,24 SAY ' Depot Labor Rate: ' GET DEP_LAB_RT
@ROW()+1,24 SAY ' Intermediate Turnover Rate: ' GET
INT_TRN_RT
@ROW()+1,24 SAY ' Depot Turnover Rate: ' GET DEP_TRN_RT
READ
*WAIT
CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ESUPSYS.PRG - EDITS SUPPLY SYSTEM DATA - 27-4-88
SET COLOR TO W/B
CLEAR
USE SUPPLY_SYS
IF EOF()
APPEND BLANK
ENDIF
SET CONFIRM ON
SET COLOR TO G/B
@4,21 TO 22,62 DOUBLE
@8,32 TO 8,51
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 3'
```

```
@ROW()+1,33 SAY 'SUPPLY SYSTEM DATA'
@ROW()+2,23 SAY ' Initial Management Cost: ' GET INI_MGT_CS
@ROW()+1,23 SAY ' Recurring Management Cost: ' GET REC_MGT_CS
@ROW()+1,23 SAY ' Base Supply Management Cost: ' GET
BS_MGT_CS
@ROW()+1,23 SAY ' Order and Ship Time CONUS: ' GET CONUS_SHIP
@ROW()+1,23 SAY ' Order and Ship Time Overseas: ' GET OS_SHIP
@ROW()+1,23 SAY ' Packing Cost CONUS: ' GET CONUS_PK_C
@ROW()+1,23 SAY ' Packing Cost Overseas: ' GET OS_PK_CST
@ROW()+1,23 SAY ' Packed Wt Ratio CONUS: ' GET CONUS_PKWT
@ROW()+1,23 SAY ' Packed Wt Ratio Overseas: ' GET OS_PKWT_RT
@ROW()+1,23 SAY ' Shipping Rate CONUS: ' GET CONUS_SHRT
@ROW()+1,23 SAY ' Shipping Rate Overseas: ' GET OS_SHIP_RT
@ROW()+1,23 SAY ' Tech Data Cost: ' GET TEC_DAT_CS
READ
*WAIT
CLOSE DATABASES
CLEAR
RETURN
Z

**********************************************************
**********************************************************

*ESESYS.PRG -
*EDITS SUPPORT EQUIPMENT DATA

SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE SE_DATA
ERASE ISERN.NDX
INDEX ON SERN TO ISERN

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@5,18 TO 19,64 DOUBLE
@9,28 TO 9,51
SET COLOR TO W/B
@7,34 SAY 'RECORD TYPE 4'
@ROW()+1,29 SAY 'SUPPORT EQUIPMENT DATA'
@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,25 SAY 'ENTER             SELECTION: '

SET COLOR TO N/W
```

```
IF ISCOLOR()
 SET COLOR TO R/B
ENDIF
@11,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,31 SAY 'HIGHLIGHTED'
@ROW(),54 GET CHOICE1 PICTURE '@A'

READ
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO ASESYS
*    ERASE SESORT.DBF
*    SORT ON SERN TO SESORT.DBF

  CASE UPPER(CHOICE1) = 'D'
CLEAR
SET COLOR TO W/B
@1,21 SAY 'RECORD TYPE 4 - SUPPORT EQUIPMENT DATA'
SET COLOR TO G/B
@2,20 TO 2,58
@4,5 TO 4,7
@ROW(),9 TO ROW(),12
@ROW(),16 TO ROW(),22
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),8 SAY ' SERN '
@ROW(),15 SAY ' SE Name '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 SERN, SE_NAME

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
```

```
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B

STORE 0 TO MSERN
@22,28 SAY 'ENTER SERN TO DELETE?'
@ROW(),50 GET MSERN PICTURE "####"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MSERN = 9999
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MSERN

*INVALID SERN ENTERED
IF EOF()
  @22,24 CLEAR TO 23,58
  @22,35 SAY 'NO SUCH SERN'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

*CONFIRM RECORD DELETION
          SET DELETED ON
          STORE .F. TO CHOICE2
          CLEAR
          SET COLOR TO G/B
          @6,20 TO 14,60 DOUBLE
          SET COLOR TO W/B
          @8,24 SAY 'Confirm Record Deletion of SERN'
          @ROW()+2,38 SAY MSERN PICTURE "####"
          @ROW()+2,32 SAY '(Y or N)?'
          @ROW(),48 GET CHOICE2 PICTURE '@! Y'
          READ

               IF CHOICE2
                DELETE
               ENDIF

     CASE UPPER(CHOICE1) = 'E'
GO TOP
CLEAR
SET COLOR TO W/B
```

```
@1,21 SAY 'RECORD TYPE 4 - SUPPORT EQUIPMENT DATA'
SET COLOR TO G/B
@2,20 TO 2,58
@4,5 TO 4,7
@ROW(),9 TO ROW(),12
@ROW(),16 TO ROW(),22
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),8 SAY ' SERN '
@ROW(),15 SAY ' SE Name '

DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 SERN, SE_NAME

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE 0 TO MSERN
@22,29 SAY 'ENTER SERN TO EDIT?'
@ROW(),49 GET MSERN PICTURE "####"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MSERN = 9999
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MSERN

*INVALID SERN ENTERED
IF EOF()
  @22,23 CLEAR TO 23,57
  @22,35 SAY 'NO SUCH SERN'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  WAIT""
  CLOSE DATABASES
  RETURN
```

```
ENDIF

*VALID SERN
STORE SE_NAME TO MSE_NAME
STORE SE_COST TO MSE_COST
STORE SE_O_M_CST TO MSE_O_M_CST
STORE NO_OF_SE TO MNO_OF_SE
STORE CURR_USAGE TO MCURR_USAGE
STORE AVAIL_TIME TO MAVAIL_TIME
STORE FACIL_COST TO MFACIL_COST

CLEAR
SET COLOR TO G/B
@5,18 TO 19,64 DOUBLE
@9,28 TO 9,51
SET COLOR TO W/B
@7,34 SAY 'RECORD TYPE 4'
@ROW()+1,29 SAY 'SUPPORT EQUIPMENT DATA'
@ROW()+2,20 SAY ' SE Resource Number (SERN) ' GET MSERN
PICTURE "####"
@ROW()+1,20 SAY ' SE Name ' GET MSE_NAME PICTURE
"XXXXXXXXXXXX"
@ROW()+1,20 SAY ' SE Cost ' GET MSE_COST PICTURE
"########.##"
@ROW()+1,20 SAY ' SE Operating and Maintenance Cost ' GET
MSE_O_M_CST PICTURE "#####.##"
@ROW()+1,20 SAY ' Number of SE ' GET MNO_OF_SE PICTURE "##"

@ROW()+1,20 SAY ' Current Usage ' GET MCURR_USAGE PICTURE
"###.##"
@ROW()+1,20 SAY ' Available SE Time ' GET MAVAIL_TIME PICTURE
"###.##"
@ROW()+1,20 SAY ' Facilities Cost 'GET MFACIL_COST PICTURE
"#######.##"
READ

REPLACE SERN WITH MSERN
REPLACE SE_NAME WITH MSE_NAME
REPLACE SE_COST WITH MSE_COST
REPLACE SE_O_M_CST WITH MSE_O_M_CST
REPLACE NO_OF_SE WITH MNO_OF_SE
REPLACE CURR_USAGE WITH MCURR_USAGE
REPLACE AVAIL_TIME WITH MAVAIL_TIME
REPLACE FACIL_COST WITH MFACIL_COST
ENDCASE

PACK
ENDDO
CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
```

```
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ASESYS.PRG -
*ADDS SUPPORT EQUIPMENT DATA RECORDS

DO WHILE .T.
APPEND BLANK
SKIP -1
CLEAR
SET COLOR TO G/B
@5,18 TO 19,65 DOUBLE
@9,29 TO 9,52
@20,18 TO 22,65 DOUBLE

SET COLOR TO W/B
@7,35 SAY 'RECORD TYPE 4'
@ROW()+1,30 SAY 'SUPPORT EQUIPMENT DATA'
SET COLOR TO W/B
@ROW()+2,20 SAY ' SE Resource Number (SERN) ' GET SERN
@ROW()+1,20 SAY ' SE Name ' GET SE_NAME
@ROW()+1,20 SAY ' SE Cost ' GET SE_COST
@ROW()+1,20 SAY ' SE Operating and Maintenance Cost ' GET
SE_O_M_CST
@ROW()+1,20 SAY ' Number of SE ' GET NO_OF_SE
@ROW()+1,20 SAY ' Current Usage ' GET CURR_USAGE
@ROW()+1,20 SAY ' Available SE Time ' GET AVAIL_TIME
@ROW()+1,20 SAY ' Facilities Cost ' GET FACIL_COST
READ

IF SERN = 0
*APPARANT BLANK RECORD
@20,18 CLEAR TO 22,65
STORE 'Y' TO ABORT
? CHR(7)
SET COLOR TO G/B
@20,18 TO 23,65 DOUBLE
SET COLOR TO W/B
@21,26 SAY '  Cannot Enter a Blank Record!  '
@22,22 SAY 'Abort new record addition? (Y or N)?'
@22,60 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
   SET CONSOLE OFF
   DELETE
   PACK
   SET CONSOLE ON
 ENDIF
```

```
ENDIF

STORE 'Y' TO ANOTHER
@20,18 CLEAR TO 23,65
SET COLOR TO G/B
@20,18 TO 22,65 DOUBLE
SET COLOR TO W/B
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
*ENDDO
ENDIF
ENDDO
SET BELL ON
RETURN
^Z

**********************************************************
**********************************************************

*ELRU.PRG -
*EDITS LINE REPLACEABLE UNIT (LRU) DATA
```

```
SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE LRU_DATA
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@4,18 TO 18,65 DOUBLE
@8,25 TO 8,58
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 5'
@ROW()+1,26 SAY 'LINE REPLACEABLE UNIT (LRU) DATA'
@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,25 SAY 'ENTER                SELECTION: '

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/B
ENDIF
@10,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,31 SAY 'HIGHLIGHTED'
@ROW(),54 GET CHOICE1 PICTURE '@A'

READ
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO ALRU

  CASE UPPER(CHOICE1) = 'D'
     CLEAR
SET COLOR TO W/B
```

```
@1,28 SAY 'RECORD TYPE 5 - LRU DATA'
SET COLOR TO G/B
@2,20 TO 2,58
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),20 SAY 'LRU Name '
GO TOP
DO WHILE .NOT. EOF()
@4,1 CLEAR TO 20,79
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),20 TO ROW(),27
@5,1
DISPLAY NEXT 15 LRU_ID, LRU_NAME

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MLRU_ID
@22,25 SAY 'ENTER LRU_ID TO DELETE?'
@ROW(),49 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,23 CLEAR TO 23,58
  @22,34 SAY 'NO SUCH LRU_ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

          SET DELETED ON
          STORE .F. TO CHOICE2
          CLEAR
          SET COLOR TO G/B
          @6,20 TO 14,60 DOUBLE
          SET COLOR TO W/B
          @8,23 SAY 'Confirm Record Deletion of LRU_ID'
          @ROW()+2,38 SAY MLRU_ID PICTURE "NNNNNNN"
          @ROW()+2,32 SAY '(Y or N)?'
          @ROW(),48 GET CHOICE2 PICTURE '@! Y'
          READ

               IF CHOICE2
                DELETE
               ENDIF

     CASE UPPER(CHOICE1) = 'E'

CLEAR
SET COLOR TO W/B
@1,28 SAY 'RECORD TYPE 5 - LRU DATA'
SET COLOR TO G/B
@2,20 TO 2,58
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),20 SAY 'LRU Name '
GO TOP
DO WHILE .NOT. EOF()
@4,1 CLEAR TO 20,79
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),20 TO ROW(),27
@5,1
DISPLAY NEXT 15 LRU_ID, LRU_NAME

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO
```

```
@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MLRU_ID
@22,26 SAY 'ENTER LRU_ID TO EDIT?'
@ROW(),49 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH LRU_ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

*VALID LRU_ID
STORE LRU_NAME TO MLRU_NAME
STORE NO_LRU_END TO MNO_LRU_END
STORE UNIT_COST TO MUNIT_COST
STORE WEIGHT TO MWEIGHT
STORE OPS_RATIO TO MOPS_RATIO
STORE DEP_RCT_CO TO MDEP_RCT_CO
STORE DEP_RCT_OS TO MDEP_RCT_OS
STORE INT_RCT_CO TO MINT_RCT_CO
STORE R_FAIL_RIP TO MR_FAIL_RIP
STORE NO_GPSE_RQ TO MNO_GPSE_RQ
STORE MTBF TO MMTBF

CLEAR
SET COLOR TO G/B
@4,18 TO 22,65 DOUBLE
@8,25 TO 8,58
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 5'
@ROW()+1,26 SAY 'LINE REPLACEABLE UNIT (LRU) DATA'
@ROW()+2,20 SAY ' LRU Identifier ' GET MLRU_ID PICTURE
"NNNNNNN"
@ROW()+1,20 SAY ' LRU Name ' GET MLRU_NAME PICTURE
"@NNNNNNNNNNNN"
@ROW()+1,20 SAY ' Number of LRUs Per End-Item ' GET
MNO_LRU_END PICTURE "##.#"
```

```
@ROW()+1,20 SAY ' Unit Cost ' GET MUNIT_COST PICTURE
"#######.##"
@ROW()+1,20 SAY ' Weight ' GET MWEIGHT PICTURE "####.#"
@ROW()+1,20 SAY ' Operating Ratio ' GET MOPS_RATIO PICTURE
"###.###"
@ROW()+1,20 SAY ' Depot Repair Cycle Time CONUS ' GET
MDEP_RCT_CO PICTURE "###.###"
@ROW()+1,20 SAY ' Depot Repair Cycle Time Overseas ' GET
MDEP_RCT_OS PICTURE "###.###"
@ROW()+1,20 SAY ' Intermediate Repair Cycle Time ' GET
MINT_RCT_CO PICTURE "###.###"
@ROW()+1,20 SAY ' Ratio Failures Repair-in-Place ' GET
MR_FAIL_RIP PICTURE "#.##"
@ROW()+1,20 SAY ' No. Kinds of GPSE Required ' GET
MNO_GPSE_RQ PICTURE "###"
@ROW()+1,20 SAY ' Mean Time Between Failure (MTBF) ' GET
MMTBF PICTURE "#######.##"
READ

REPLACE LRU_ID WITH MLRU_ID
REPLACE LRU_NAME WITH MLRU_NAME
REPLACE NO_LRU_END WITH MNO_LRU_END
REPLACE UNIT_COST WITH MUNIT_COST
REPLACE WEIGHT WITH MWEIGHT
REPLACE OPS_RATIO WITH MOPS_RATIO
REPLACE DEP_RCT_CO WITH MDEP_RCT_CO
REPLACE DEP_RCT_OS WITH MDEP_RCT_OS
REPLACE INT_RCT_CO WITH MINT_RCT_CO
REPLACE R_FAIL_RIP WITH MR_FAIL_RIP
REPLACE NO_GPSE_RQ WITH MNO_GPSE_RQ
REPLACE MTBF WITH MMTBF
ENDCASE

PACK
ENDDO
CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ALRU.PRG -
*ADDS LRU DATA RECORDS

DO WHILE .T.
APPEND BLANK
SKIP -1
```

```
CLEAR
SET COLOR TO G/B
@1,18 TO 19,65 DOUBLE
@5,25 TO 5,58
@20,18 TO 22,65 DOUBLE

SET COLOR TO W/B
@3,36 SAY 'RECORD TYPE 5'
@ROW()+1,26 SAY 'LINE REPLACEABLE UNIT (LRU) DATA'

@ROW()+2,20 SAY ' LRU Identifier ' GET LRU_ID
@ROW()+1,20 SAY ' LRU Name ' GET LRU_NAME
@ROW()+1,20 SAY ' Number of LRUs Per End-Item ' GET
NO_LRU_END
@ROW()+1,20 SAY ' Unit Cost ' GET UNIT_COST
@ROW()+1,20 SAY ' Weight ' GET WEIGHT
@ROW()+1,20 SAY ' Operating Ratio ' GET OPS_RATIO
@ROW()+1,20 SAY ' Depot Repair Cycle Time CONUS ' GET
DEP_RCT_CO
@ROW()+1,20 SAY ' Depot Repair Cycle Time Overseas ' GET
DEP_RCT_OS
@ROW()+1,20 SAY ' Intermediate Repair Cycle Time ' GET
INT_RCT_CO
@ROW()+1,20 SAY ' Ratio Failures Repair-in-Place ' GET
R_FAIL_RIP
@ROW()+1,20 SAY ' No. Kinds of GPSE Required ' GET NO_GPSE_RQ

@ROW()+1,20 SAY ' Mean Time Between Failure (MTBF) ' GET MTBF

READ

IF LRU_ID = "         "
*APPARANT BLANK RECORD
@20,18 CLEAR TO 22,65
STORE 'Y' TO ABORT
? CHR(7)
SET COLOR TO G/B
@20,18 TO 23,65 DOUBLE
SET COLOR TO W/B
@21,25 SAY '  Cannot Enter a Blank Record!  '
@22,21 SAY 'Abort new record addition? (Y or N)?'
@22,59 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
    SET CONSOLE OFF
    DELETE
    PACK
    SET CONSOLE ON
 ENDIF
ENDIF

STORE 'Y' TO ANOTHER
```

```
@20,18 CLEAR TO 23,65
SET COLOR TO G/B
@20,18 TO 22,65 DOUBLE
SET COLOR TO W/B
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
*ENDDO
ENDIF
ENDDO
SET BELL ON
RETURN
^Z

**********************************************************
**********************************************************

*ELRUFM.PRG -
*EDITS LRU FAILURE MODE DATA

SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE LRU_FM_D
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@4,18 TO 18,65 DOUBLE
@8,29 TO 8,51
SET COLOR TO W/B
@6,34 SAY 'RECORD TYPE 6'
@ROW()+1,30 SAY 'LRU FAILURE MODE DATA'

@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,25 SAY 'ENTER             SELECTION: '

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/B
ENDIF
@10,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,31 SAY 'HIGHLIGHTED'
@ROW(),54 GET CHOICE1 PICTURE '@A'

READ
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO ALRUFM

  CASE UPPER(CHOICE1) = 'D'
     CLEAR
SET COLOR TO W/B
@1,22 SAY 'RECORD TYPE 6 - LRU FAILURE MODE DATA'
SET COLOR TO G/B
```

```
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),17 TO ROW(),19
@ROW(),21 TO ROW(),28
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW(),17 SAY 'FM'
@ROW(),21 SAY 'FM OR'
@ROW()+1,5 SAY 'NO.'
@ROW(),17 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),21 SAY 'SRU Name '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_ID, FM_IDEN_NO, FM_SRU_NAM

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B

STORE SPACE(7) TO MLRU_ID
@22,25 SAY 'ENTER LRU ID TO DELETE?'
@ROW(),50 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH LRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
```

```
    WAIT""
    CLOSE DATABASES
    RETURN
  ENDIF

  STORE 0 TO MFM_IDEN_NO
  @22,24 CLEAR TO 23,57
  @23,28 SAY 'ENTER FAILURE MODE NO. '
  @ROW(),52 GET MFM_IDEN_NO PICTURE "##"
  READ
  LOCATE FOR FM_IDEN_NO = MFM_IDEN_NO .AND. LRU_ID = MLRU_ID


  *INVALID LRU_ID .AND. FM_IDEN_NO ENTERED
  IF EOF()
    @22,24 CLEAR TO 23,57
    @22,34 SAY 'NO SUCH RECORD'
    @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
    ?? CHR(7)
    WAIT""
    CLOSE DATABASES
    RETURN
  ENDIF

  SET DELETED ON
  STORE .F. TO CHOICE2
  CLEAR
  SET COLOR TO G/B
  @6,20 TO 15,60 DOUBLE
  SET COLOR TO W/B
  @8,27 SAY 'Confirm Record Deletion of'
  @ROW()+2,33 SAY 'LRU ID: '
  @ROW(),42 SAY MLRU_ID PICTURE "NNNNNNN"
  @ROW()+1,35 SAY 'FM NO.: '
  @ROW(),44 SAY MFM_IDEN_NO PICTURE "##"
  @ROW()+2,35 SAY '(Y or N)?'
  @ROW(),45 GET CHOICE2 PICTURE '@! Y'
  READ

  IF CHOICE2
   DELETE
  ENDIF

       CASE UPPER(CHOICE1) = 'E'
  CLEAR
  SET COLOR TO W/B
  @1,22 SAY 'RECORD TYPE 6 - LRU FAILURE MODE DATA'
  SET COLOR TO G/B
  @4,5 TO 4,7
  @ROW(),9 TO ROW(),14
  @ROW(),17 TO ROW(),19
  @ROW(),21 TO ROW(),28
```

```
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW(),17 SAY 'FM'
@ROW(),21 SAY 'FM OR'
@ROW()+1,5 SAY 'NO.'
@ROW(),17 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),21 SAY 'SRU Name '
GO TOP
DC WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_ID, FM_IDEN_NO, FM_SRU_NAM

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MLRU_ID
@22,26 SAY 'ENTER LRU ID TO EDIT?'
@ROW(),48 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH LRU_ID'
  @ROW()+1,24 SAY 'PRESS A KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF
```

```
STORE 0 TO MFM_IDEN_NO
@22,24 CLEAR TO 23,57
@23,28 SAY 'ENTER FAILURE MODE NO. '
@ROW(),52 GET MFM_IDEN_NO PICTURE "##"
READ
LOCATE FOR FM_IDEN_NO = MFM_IDEN_NO .AND. LRU_ID = MLRU_ID


*INVALID LRU_ID .AND. FM_IDEN_NO ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH RECORD'
  @ROW()+1,24 SAY 'PRESS A KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

*VALID LRU_ID
STORE FM_IDEN_NO TO MFM_IDEN_NO
STORE FM_RATIO TO MFM_RATIO
STORE SRU_ID TO MSRU_ID
STORE FM_SRU_NAM TO MFM_SRU_NAM
STORE NO_NEW_PRT TO MNO_NEW_PRT
STORE NO_STD_PRT TO MNO_STD_PRT
STORE REP_PRT_CS TO MREP_PRT_CS
STORE WT_PC_PART TO MWT_PC_PART
STORE NO_TRN_DEP TO MNO_TRN_DEP
STORE NO_TRN_INT TO MNO_TRN_INT
STORE DEP_MX_MHR TO MDEP_MX_MHR
STORE INT_MX_MHR TO MINT_MX_MHR
STORE WKS_MX_TRN TO MWKS_MX_TRN
STORE TRNG_CST TO MTRNG_CST
STORE TEC_DT_PGS TO MTEC_DT_PGS
STORE NO_SPSE_RQ TO MNO_SPSE_RQ
STORE FRC_LRU_FM TO MFRC_LRU_FM
STORE SE_HRS_REP TO MSE_HRS_REP

CLEAR
SET COLOR TO G/B
@0,18 TO 23,65 DOUBLE
@3,25 TO 3,58
SET COLOR TO W/B
@1,36 SAY 'RECORD TYPE 6'
@ROW()+1,32 SAY 'LRU FAILURE MODE DATA'
@ROW()+2,20 SAY ' LRU Identifier: ' GET MLRU_ID PICTURE
"NNNNNNN"
@ROW()+1,20 SAY ' Failure Mode Identifier Number: ' GET
MFM_IDEN_NO PICTURE "##"
@ROW()+1,20 SAY ' Failure Mode Ratio: ' GET MFM_RATIO PICTURE
"#.###"
```

```
@ROW()+1,20 SAY ' SRU Identifier: ' GET MSRU_ID PICTURE
"NNNNNNN"
@ROW()+1,20 SAY ' FM or SRU Name: ' GET MFM_SRU_NAM PICTURE
"NNNNNNNN"
@ROW()+1,20 SAY ' No. New Parts: ' GET MNO_NEW_PRT PICTURE
"###"
@ROW()+1,20 SAY ' No. Standard Parts: ' GET MNO_STD_PRT
PICTURE "###"
@ROW()+1,20 SAY ' Repair Parts Cost: ' GET MREP_PRT_CS
PICTURE "#######.##"
@ROW()+1,20 SAY ' Weight of Piece Parts: ' GET MWT_PC_PART
PICTURE "####.##"
@ROW()+1,20 SAY ' No. Persons Trained at Depot: ' GET
MNO_TRN_DEP PICTURE "###"
@ROW()+1,20 SAY ' No. Persons Trained at Intermediate: ' GET
MNO_TRN_INT PICTURE "###"
@ROW()+1,20 SAY ' Depot Maintenance Manhours: ' GET
MDEP_MX_MHR PICTURE "###.###"
@ROW()+1,20 SAY ' Intermediate Maintenance Manhours: ' GET
MINT_MX_MHR PICTURE "###.###"
@ROW()+1,20 SAY ' Weeks Maintenance Training: ' GET
MWKS_MX_TRN PICTURE "###.###"
@ROW()+1,20 SAY ' Training Cost: ' GET MTRNG_CST PICTURE
"#####.##"
@ROW()+1,20 SAY ' Technical Data Pages: ' GET MTEC_DT_PGS
PICTURE "###.##"
@ROW()+1,20 SAY ' No. SPSE Required: ' GET MNO_SPSE_RQ
PICTURE "###"
@ROW()+1,20 SAY ' Forced LRU Failure Mode Decisions: ' GET
MFRC_LRU_FM PICTURE "###"
@ROW()+1,20 SAY ' SE Hours per Repair: ' GET MSE_HRS_REP
PICTURE "###.###"

READ

REPLACE LRU_ID WITH MLRU_ID
REPLACE FM_IDEN_NO WITH MFM_IDEN_NO
REPLACE FM_RATIO WITH MFM_RATIO
REPLACE SRU_ID WITH MSRU_ID
REPLACE FM_SRU_NAM WITH MFM_SRU_NAM
REPLACE NO_NEW_PRT WITH MNO_NEW_PRT
REPLACE NO_STD_PRT WITH MNO_STD_PRT
REPLACE REP_PRT_CS WITH MREP_PRT_CS
REPLACE WT_PC_PART WITH MWT_PC_PART
REPLACE NO_TRN_DEP WITH MNO_TRN_DEP
REPLACE NO_TRN_INT WITH MNO_TRN_INT
REPLACE DEP_MX_MHR WITH MDEP_MX_MHR
REPLACE INT_MX_MHR WITH MINT_MX_MHR
REPLACE WKS_MX_TRN WITH MWKS_MX_TRN
REPLACE TRNG_CST WITH MTRNG_CST
REPLACE TEC_DT_PGS WITH MTEC_DT_PGS
REPLACE NO_SPSE_RQ WITH MNO_SPSE_RQ
REPLACE FRC_LRU_FM WITH MFRC_LRU_FM
REPLACE SE_HRS_REP WITH MSE_HRS_REP

ENDCASE

PACK
ENDDO
CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ALRUFM.PRG -
*ADDS LRU FAILURE MODE DATA RECORDS

DO WHILE .T.
APPEND BLANK
SKIP -1
CLEAR
SET COLOR TO G/B
@0,18 TO 23,65 DOUBLE
@3,29 TO 3,51
SET COLOR TO W/B
@1,34 SAY 'RECORD TYPE 6'
@2,30 SAY 'LRU FAILURE MODE DATA'

@ROW()+2,20 SAY ' LRU Identifier: ' GET LRU_ID
@ROW()+1,20 SAY ' Failure Mode Identifier Number: ' GET
FM_IDEN_NO
@ROW()+1,20 SAY ' Failure Mode Ratio: ' GET  FM_RATIO
@ROW()+1,20 SAY ' SRU Identifier: ' GET SRU_ID
@ROW()+1,20 SAY ' FM or SRU Name: ' GET FM_SRU_NAM
@ROW()+1,20 SAY ' No. New Parts: ' GET NO_NEW_PRT
@ROW()+1,20 SAY ' No. Standard Parts: ' GET NO_STD_PRT

@ROW()+1,20 SAY ' Repair Parts Cost: ' GET REP_PRT_CS
@ROW()+1,20 SAY ' Weight of Piece Parts: ' GET WT_PC_PART

@ROW()+1,20 SAY ' No. Persons Trained at Depot: ' GET
NO_TRN_DEP
@ROW()+1,20 SAY ' No. Persons Trained at Intermediate: ' GET
NO_TRN_INT
@ROW()+1,20 SAY ' Depot Maintenance Manhours: ' GET
DEP_MX_MHR
@ROW()+1,20 SAY ' Intermediate Maintenance Manhours: ' GET
INT_MX_MHR
```

```
@ROW()+1,20 SAY ' Weeks Maintenance Training: ' GET
WKS_MX_TRN
@ROW()+1,20 SAY ' Training Cost: ' GET TRNG_CST
@ROW()+1,20 SAY ' Technical Data Pages: ' GET TEC_DT_PGS

@ROW()+1,20 SAY ' No. SPSE Required: ' GET NO_SPSE_RQ
@ROW()+1,20 SAY ' Forced LRU Failure Mode Decisions: ' GET
FRC_LRU_FM
@ROW()+1,20 SAY ' SE Hours per Repair: ' GET SE_HRS_REP
READ

IF LRU_ID = "       "
*APPARANT BLANK RECORD
@20,18 CLEAR TO 22,65
STORE 'Y' TO ABORT
? CHR(7)
SET COLOR TO G/B
@20,18 TO 23,65 DOUBLE
SET COLOR TO W/B
@21,25 SAY '  Cannot Enter a Blank Record!  '
@22,21 SAY 'Abort new record addition? (Y or N)?'
@22,59 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
   SET CONSOLE OFF
   DELETE
   PACK
   SET CONSOLE ON
 ENDIF
ENDIF

STORE 'Y' TO ANOTHER
@20,18 CLEAR TO 23,65
SET COLOR TO G/B
@20,18 TO 22,65 DOUBLE
SET COLOR TO W/B
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
```

```
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
*ENDDO
ENDIF
ENDDO

SET BELL ON
RETURN
^Z

**********************************************************
**********************************************************

*ESRU.PRG -
*EDITS SHOP REPLACEABLE UNIT (SRU) DATA

SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE SRU_DATA
ERASE ISID.NDX
INDEX ON SRU_ID TO ISID

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@4,18 TO 18,65 DOUBLE
@8,25 TO 8,58
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 7'
@ROW()+1,26 SAY 'SHOP REPLACEABLE UNIT (SRU) DATA'
@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
```

```
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,26 SAY 'ENTER             SELECTION: '

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/B
ENDIF
@10,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,32 SAY 'HIGHLIGHTED'
@ROW(),55 GET CHOICE1 PICTURE '@A'

READ
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO ASRU

  CASE UPPER(CHOICE1) = 'D'
     CLEAR
SET COLOR TO W/B
@1,28 SAY 'RECORD TYPE 7 - SRU DATA'
SET COLOR TO G/B
@2,20 TO 2,58
@4,5 TO 4,7

@ROW(),9 TO ROW(),14
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'SRU ID '
*@ROW(),17 SAY 'SRU Name '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 SRU_ID
STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ
IF UPPER(NEXT15) = 'N'
```

```
  GO BOTTOM
  SKIP
ENDIF
ENDDO
@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MSRU_ID
@22,25 SAY 'ENTER SRU ID TO DELETE?'
@ROW(),50 GET MSRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ
IF MSRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MSRU_ID

*INVALID SRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH SRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
 WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

          SET DELETED ON
          STORE .F. TO CHOICE2

          CLEAR
          SET COLOR TO G/B
          @6,20 TO 14,60 DOUBLE
          SET COLOR TO W/B
          @8,23 SAY 'Confirm Record Deletion of SRU_ID'

          @ROW()+2,38 SAY MSRU_ID PICTURE "NNNNNNN"
          @ROW()+2,32 SAY '(Y or N)?'
          @ROW(),48 GET CHOICE2 PICTURE '@! Y'
          READ

               IF CHOICE2
                DELETE
               ENDIF

     CASE UPPER(CHOICE1) = 'E'
CLEAR
SET COLOR TO W/B
@1,28 SAY 'RECORD TYPE 7 - SRU DATA'
SET COLOR TO G/B
```

```
@2,20 TO 2,58
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
*ROW(),16 TO ROW(),22
@21,22 TO 24,59 DOUBLE
SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'SRU ID '
*@ROW(),15 SAY ' SE Name '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 SRU_ID
STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ
IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MSRU_ID
@22,26 SAY 'ENTER SRU ID TO EDIT?'
@ROW(),49 GET MSRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MSRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MSRU_ID

*INVALID SRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH SRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

*VALID SRU_ID
```

```
STORE SRU_CST TO MSRU_CST
STORE SRU_WEIGHT TO MSRU_WEIGHT
STORE CST_PP_REP TO MCST_PP_REP
STORE WT_PP_ASSY TO MWT_PP_ASSY
STORE NO_NEW_PP TO MNO_NEW_PP
STORE NO_STD_PP TO MNO_STD_PP
STORE NO_PGS_T_D TO MNO_PGS_T_D
STORE NO_KIND_SE TO MNO_KIND_SE
STORE DEP_RCT_CO TO MDEP_RCT_CO
STORE DEP_RCT_OS TO MDEP_RCT_OS
STORE INT_RCT TO MINT_RCT
STORE DEP_MX_MHR TO MDEP_MX_MHR
STORE INT_MX_MHR TO MINT_MX_MHR
STORE DEP_TRNEES TO MDEP_TRNEES
STORE INT_TRNEES TO MINT_TRNEES
STORE MX_TRN_WKS TO MMX_TRN_WKS
STORE MX_TRN_CST TO MMX_TRN_CST
STORE FRC_SRU_DE TO MFRC_SRU_DE
STORE SE_HRS_REP TO MSE_HRS_REP
CLEAR
SET COLOR TO G/B
@0,13 TO 23,71 DOUBLE
@3,24 TO 3,57
SET COLOR TO W/B
@1,35 SAY 'RECORD TYPE 7'
@2,25 SAY 'SHOP REPLACEABLE UNIT (SRU) DATA'
@ROW()+2,15 SAY ' SRU Identifier: ' GET MSRU_ID PICTURE
"NNNNNNN"
@ROW(),46 SAY ' SRU Cost: ' GET MSRU_CST PICTURE "#####.##"
@ROW()+1,15 SAY ' Weight of SRU: ' GET MSRU_WEIGHT PICTURE
"####.##"
@ROW()+1,15 SAY ' Cost of Piece Parts and Assys Per SRU
Repair: ' GET MCST_PP_REP PICTURE "#####.##"
@ROW()+1,16 SAY 'Weight of Parts and Assemblies: ' GET
MWT_PP_ASSY PICTURE "####.##"
@ROW()+1,15 SAY ' No. of New Piece Parts and Assemblies Used:
' GET MNO_NEW_PP PICTURE "###"
@ROW()+1,15 SAY ' No. Standard Piece Parts per Repair: ' GET
MNO_STD_PP PICTURE "###"
@ROW()+1,15 SAY ' No. Pages of Tech Data per Repair: ' GET
MNO_PGS_T_D PICTURE "###.##"
@ROW()+1,15 SAY ' No. Kinds of SE Used per SRU Repair: ' GET
MNO_KIND_SE PICTURE "##"
@ROW()+1,15 SAY ' Depot Repair Cycle Time CONUS: ' GET
MDEP_RCT_CO PICTURE "###.###"
@ROW()+1,15 SAY ' Depot Repair Cycle Time Overseas: ' GET
MDEP_RCT_OS PICTURE "###.###"
@ROW()+1,15 SAY ' Intermediate Repair Cycle Time: ' GET
MINT_RCT PICTURE "###.###"
@ROW()+1,15 SAY ' Depot Maintenance Manhours per Repair: '
GET MDEP_MX_MHR PICTURE "###.###"
@ROW()+1,15 SAY ' Intermediate Maintenance Manhours per
```

```
Repair: ' GET MINT_MX_MHR PICTURE "###.###"
@ROW()+1,15 SAY ' Depot Trainees: ' GET MDEP_TRNEES PICTURE
"###"
@ROW()+1,15 SAY ' Intermediate Trainees: ' GET MINT_TRNEES
PICTURE "###"
@ROW()+1,15 SAY ' Maintenance Training Weeks: ' GET
MMX_TRN_WKS PICTURE "###.###"
@ROW()+1,15 SAY ' Maintenance Training Cost: ' GET
MMX_TRN_CST PICTURE "#####.##"
@ROW()+1,15 SAY ' Forced SRU Decisions: ' GET MFRC_SRU_DE
PICTURE "###"
@ROW()+1,15 SAY ' SE Hours per Repair: ' GET MSE_HRS_REP
PICTURE "###.###"
READ
REPLACE SRU_ID WITH MSRU_ID
REPLACE SRU_CST WITH MSRU_CST
REPLACE SRU_WEIGHT WITH MSRU_WEIGHT
REPLACE CST_PP_REP WITH MCST_PP_REP
REPLACE WT_PP_ASSY WITH MWT_PP_ASSY
REPLACE NO_NEW_PP WITH MNO_NEW_PP
REPLACE NO_STD_PP WITH MNO_STD_PP
REPLACE NO_PGS_T_D WITH MNO_PGS_T_D
REPLACE NO_KIND_SE WITH MNO_KIND_SE
REPLACE DEP_RCT_CO WITH MDEP_RCT_CO
REPLACE DEP_RCT_OS WITH MDEP_RCT_OS
REPLACE INT_RCT WITH MINT_RCT
REPLACE DEP_MX_MHR WITH MDEP_MX_MHR
REPLACE INT_MX_MHR WITH MINT_MX_MHR
REPLACE DEP_TRNEES WITH MDEP_TRNEES
REPLACE INT_TRNEES WITH MINT_TRNEES
REPLACE MX_TRN_WKS WITH MMX_TRN_WKS
REPLACE MX_TRN_CST WITH MMX_TRN_CST
REPLACE FRC_SRU_DE WITH MFRC_SRU_DE
REPLACE SE_HRS_REP WITH MSE_HRS_REP
ENDCASE
PACK
ENDDO

CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ASRU.PRG -
*ADDS SRU DATA RECORDS
```

```
DO WHILE .T.
APPEND BLANK
SKIP -1
CLEAR
SET COLOR TO G/B
@0,13 TO 23,71 DOUBLE
@3,24 TO 3,57

SET COLOR TO W/B
@1,35 SAY 'RECORD TYPE 7'
@2,25 SAY 'SHOP REPLACEABLE UNIT (SRU) DATA'
@ROW()+2,15 SAY ' SRU Identifier: ' GET SRU_ID
@ROW(),46 SAY ' SRU Cost: ' GET SRU_CST
@ROW()+1,15 SAY ' Weight of SRU: ' GET SRU_WEIGHT
@ROW()+1,15 SAY ' Cost of Piece Parts and Assys Per SRU
Repair: ' GET CST_PP_REP
@ROW()+1,16 SAY 'Weight of Parts and Assemblies: ' GET
WT_PP_ASSY
@ROW()+1,15 SAY ' No. of New Piece Parts and Assemblies Used:
' GET NO_NEW_PP
@ROW()+1,15 SAY ' No. Standard Piece Parts per Repair: ' GET
NO_STD_PP
@ROW()+1,15 SAY ' No. Pages of Tech Data per Repair: ' GET
NO_PGS_T_D
@ROW()+1,15 SAY ' No. Kinds of SE Used per SRU Repair: ' GET
NO_KIND_SE
@ROW()+1,15 SAY ' Depot Repair Cycle Time CONUS: ' GET
DEP_RCT_CO
@ROW()+1,15 SAY ' Depot Repair Cycle Time Overseas: ' GET
DEP_RCT_OS
@ROW()+1,15 SAY ' Intermediate Repair Cycle Time: ' GET
INT_RCT
@ROW()+1,15 SAY ' Depot Maintenance Manhours per Repair: '
GET DEP_MX_MHR
@ROW()+1,15 SAY ' Intermediate Maintenance Manhours per
Repair: ' GET INT_MX_MHR
@ROW()+1,15 SAY ' Depot Trainees: ' GET DEP_TRNEES
@ROW()+1,15 SAY ' Intermediate Trainees: ' GET INT_TRNEES
@ROW()+1,15 SAY ' Maintenance Training Weeks: ' GET
MX_TRN_WKS
@ROW()+1,15 SAY ' Maintenance Training Cost: ' GET MX_TRN_CST
@ROW()+1,15 SAY ' Forced SRU Decisions: ' GET FRC_SRU_DE
@ROW()+1,15 SAY ' SE Hours per Repair: ' GET SE_HRS_REP
READ

IF SRU_ID = "        "
*APPARANT BLANK RECORD
@20,13 CLEAR TO 23,71
STORE 'Y' TO ABORT
? CHR(7)

SET COLOR TO G/B
```

```
@20,13 TO 23,71 DOUBLE
SET COLOR TO W/B
@21,25 SAY '  Cannot Enter a Blank Record!   '
@22,21 SAY 'Abort new record addition? (Y or N)?'
@22,59 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
   SET CONSOLE OFF
   DELETE
   PACK
   SET CONSOLE ON
 ENDIF
ENDIF

STORE 'Y' TO ANOTHER
@20,13 CLEAR TO 23,71
SET COLOR TO G/B
@20,13 TO 22,71 DOUBLE
SET COLOR TO W/B
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
```

```
*ENDDO
ENDIF

ENDDO

SET BELL ON
RETURN
^Z

**********************************************************
**********************************************************

*ECROSS.PRG -
*EDITS LRU/SRU/SE CROSS REFERENCE DATA

SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE CROSS_RE
ERASE ILSID.NDX
INDEX ON LRU_SRU_ID TO ILSID

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@4,18 TO 18,65 DOUBLE
@8,25 TO 8,58
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 8'
@ROW()+1,26 SAY 'LRU/SRU/SE CROSS REFERENCE DATA'

@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,26 SAY 'ENTER             SELECTION: '

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/B
ENDIF
@10,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,32 SAY 'HIGHLIGHTED'
@ROW(),55 GET CHOICE1 PICTURE '@A'

READ
```

```
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO ACROSS

  CASE UPPER(CHOICE1) = 'D'
     CLEAR
SET COLOR TO W/B
@1,21 SAY 'RECORD TYPE 8 - CROSS REFERENCE DATA'
SET COLOR TO G/B
@2,20 TO 2,57
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),17 TO ROW(),21
@21,22 TO 24,59 DOUBLE

SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW(),9 SAY 'LRU OR'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'SRU ID '
@ROW(),17 SAY 'FM ID '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_SRU_ID, FM_ID_NO

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO.

@22,23 CLEAR TO 23,57
SET COLOR TO W/B

STORE SPACE(7) TO MLRU_SRU_ID
@22,23 SAY 'ENTER LRU/SRU ID TO DELETE?'
@ROW(),51 GET MLRU_SRU_ID PICTURE "NNNNNNN"
```

```
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_SRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_SRU_ID

*INVALID LRU_SRU_ID ENTERED
IF EOF()
  @22,23 CLEAR TO 23,57
  @22,32 SAY 'NO SUCH LRU/SRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

STORE 0 TO MFM_ID_NO
@22,23 CLEAR TO 23,57
@23,28 SAY 'ENTER FAILURE MODE NO. '
@ROW(),52 GET MFM_ID_NO PICTURE "##"
READ
LOCATE FOR FM_ID_NO = MFM_ID_NO .AND. LRU_SRU_ID =
MLRU_SRU_ID

*INVALID LRU_ID .AND. FM_IDEN_NO ENTERED
IF EOF()
  @22,23 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH RECORD'
  @ROW()+1,24 SAY 'PRESS A KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF


SET DELETED ON
STORE .F. TO CHOICE2
CLEAR
SET COLOR TO G/B
@6,20 TO 15,60 DOUBLE
SET COLOR TO W/B
@8,27 SAY 'Confirm Record Deletion of'
@ROW()+2,31 SAY 'LRU/SRU ID: '
@ROW(),44 SAY MLRU_SRU_ID PICTURE "NNNNNNN"
@ROW()+1,35 SAY 'FM NO.: '
@ROW(),44 SAY MFM_ID_NO PICTURE "##"
```

```
@ROW()+2,35 SAY '(Y or N)?'
@ROW(),45 GET CHOICE2 PICTURE '@! Y'
READ

IF CHOICE2
 DELETE
ENDIF

CASE UPPER(CHOICE1) = 'E'
CLEAR
SET COLOR TO W/B
@1,21 SAY 'RECORD TYPE 8 - CROSS REFERENCE DATA'
SET COLOR TO G/B
@2,20 TO 2,57
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),17 TO ROW(),21
@21,22 TO 24,59 DOUBLE

SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW(),9 SAY 'LRU OR'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'SRU ID '
@ROW(),17 SAY 'FM ID '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_SRU_ID, FM_ID_NO

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B

STORE SPACE(7) TO MLRU_SRU_ID
@22,24 SAY 'ENTER LRU/SRU ID TO EDIT?'
@ROW(),51 GET MLRU_SRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_SRU_ID = "9999"
```

```
   CLEAR
   CLOSE DATABASES
   RETURN
ENDIF
SEEK MLRU_SRU_ID

*INVALID LRU_SRU_ID ENTERED
IF EOF()
   @22,24 CLEAR TO 23,57
   @22,32 SAY 'NO SUCH LRU/SRU ID'
   @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'

   ?? CHR(7)
   WAIT""
   CLOSE DATABASES
   RETURN
ENDIF

STORE 0 TO MFM_ID_NO
@22,24 CLEAR TO 23,57
@23,28 SAY 'ENTER FAILURE MODE NO. '
@ROW(),52 GET MFM_ID_NO PICTURE "##"
READ
LOCATE FOR FM_ID_NO = MFM_ID_NO .AND. LRU_SRU_ID =
MLRU_SRU_ID

*INVALID LRU_ID .AND. FM_IDEN_NO ENTERED
IF EOF()
   @22,24 CLEAR TO 23,57
   @22,34 SAY 'NO SUCH RECORD'
   @ROW()+1,24 SAY 'PRESS A KEY TO RETURN TO EDIT MENU'
   ?? CHR(7)
   WAIT""
   CLOSE DATABASES
   RETURN
ENDIF

*VALID LRU_SRU_ID
STORE REC_TYP_ID TO MREC_TYP_ID
STORE LRU_SRU_ID TO MLRU_SRU_ID
STORE FM_ID_NO TO MFM_ID_NO
STORE SE_RES_NO1 TO MSE_RES_01
STORE SE_RES_NO2 TO MSE_RES_02
STORE SE_RES_NO3 TO MSE_RES_03
STORE SE_RES_NO4 TO MSE_RES_04
STORE SE_RES_NO5 TO MSE_RES_05
STORE SE_RES_NO6 TO MSE_RES_06
STORE SE_RES_NO7 TO MSE_RES_07
STORE SE_RES_NO8 TO MSE_RES_08
STORE SE_RES_NO9 TO MSE_RES_09
STORE SE_RES_N10 TO MSE_RES_10
STORE SE_RES_N11 TO MSE_RES_11
STORE SE_RES_N12 TO MSE_RES_12
STORE SE_RES_N13 TO MSE_RES_13
STORE SE_RES_N14 TO MSE_RES_14
STORE SE_RES_N15 TO MSE_RES_15
STORE SE_RES_N16 TO MSE_RES_16
STORE CONTIN_IND TO MCONTIN_IND

CLEAR
SET COLOR TO G/B
@4,10 TO 22,72 DOUBLE
@8,25 to 8,57

SET COLOR TO W/B
@6,35 SAY 'RECORD TYPE 8'
@ROW()+1,26 SAY 'LRU/SRU/SE CROSS REFERENCE DATA'
@ROW()+2,12 SAY ' Record Type Identifier: ' GET MREC_TYP_ID
PICTURE "#"
@ROW()+1,12 SAY ' LRU or SRU Identifier: ' GET MLRU_SRU_ID
PICTURE "NNNNNNN"
@ROW()+1,12 SAY ' Failure Mode Identification Number: ' GET
MFM_ID_NO PICTURE "##"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_01
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_02
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_03
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_04
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_05
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_06
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_07
PICTURE "####"
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET MSE_RES_08
PICTURE "####"
@ROW()-7,43 SAY ' SE Resource Numbers: ' GET MSE_RES_09
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_10
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_11
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_12
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_13
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_14
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_15
PICTURE "####"
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET MSE_RES_16
```

```
PICTURE "####"
@ROW()+1,12 SAY ' Continuation Indicator: ' GET MCONTIN_IND
PICTURE "##"
READ

REPLACE REC_TYP_ID WITH MREC_TYP_ID
REPLACE LRU_SRU_ID WITH MLRU_SRU_ID
REPLACE FM_ID_NO WITH MFM_ID_NO
REPLACE SE_RES_NO1 WITH MSE_RES_01
REPLACE SE_RES_NO2 WITH MSE_RES_02
REPLACE SE_RES_NO3 WITH MSE_RES_03
REPLACE SE_RES_NO4 WITH MSE_RES_04
REPLACE SE_RES_NO5 WITH MSE_RES_05
REPLACE SE_RES_NO6 WITH MSE_RES_06
REPLACE SE_RES_NO7 WITH MSE_RES_07
REPLACE SE_RES_NO8 WITH MSE_RES_08
REPLACE SE_RES_NO9 WITH MSE_RES_09
REPLACE SE_RES_N10 WITH MSE_RES_10
REPLACE SE_RES_N11 WITH MSE_RES_11
REPLACE SE_RES_N12 WITH MSE_RES_12
REPLACE SE_RES_N13 WITH MSE_RES_13
REPLACE SE_RES_N14 WITH MSE_RES_14
REPLACE SE_RES_N15 WITH MSE_RES_15
REPLACE SE_RES_N16 WITH MSE_RES_16
REPLACE CONTIN_IND WITH MCONTIN_IND
         ENDCASE

PACK
ENDDO
CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*ACROSS.PRG -
*ADDS LRU/SRU/SE DATA RECORDS

DO WHILE .T.
APPEND BLANK
SKIP -1
CLEAR
SET COLOR TO G/B
@4,10 TO 22,72 DOUBLE
@8,25 to 8,57

SET COLOR TO W/B
```

```
@6,35 SAY 'RECORD TYPE 8'
@ROW()+1,26 SAY 'LRU/SRU/SE CROSS REFERENCE DATA'
@ROW()+2,12 SAY ' Record Type Identifier: ' GET REC_TYP_ID
@ROW()+1,12 SAY ' LRU or SRU Identifier: ' GET LRU_SRU_ID
@ROW()+1,12 SAY ' Failure Mode Identification Number: ' GET
FM_ID_NO
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO1
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO2
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO3
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO4
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO5
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO6
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO7
@ROW()+1,12 SAY ' SE Resource Numbers: ' GET SE_RES_NO8
@ROW()-7,43 SAY ' SE Resource Numbers: ' GET SE_RES_NO9
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N10
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N11
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N12
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N13
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N14
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N15
@ROW()+1,43 SAY ' SE Resource Numbers: ' GET SE_RES_N16
@ROW()+1,12 SAY ' Continuation Indicator: ' GET CONTIN_IND
READ

IF LRU_SRU_ID = "       "
*APPARANT BLANK RECORD
@20,10 CLEAR TO 22,72
STORE 'Y' TO ABORT
? CHR(7)
SET COLOR TO G/B
@20,10 TO 23,72 DOUBLE
SET COLOR TO W/B
@21,25 SAY '  Cannot Enter a Blank Record!  '
@22,21 SAY 'Abort new record addition? (Y or N)?'
@22,59 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
   SET CONSOLE OFF
   DELETE
   PACK
   SET CONSOLE ON
 ENDIF
ENDIF

STORE 'Y' TO ANOTHER
@20,10 CLEAR TO 23,72
SET COLOR TO G/B
@20,10 TO 22,72 DOUBLE
SET COLOR TO W/B
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
```

```
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
*ENDDO
ENDIF

ENDDO

SET BELL ON
RETURN
^Z

***********************************************************
***********************************************************

*EOUTOPT.PRG - EDITS OUTPUT OPTIONS DATA
SET CONFIRM ON
SET COLOR TO W/B
CLEAR
USE OUTP_OPT
IF EOF()
APPEND BLANK
ENDIF
SET COLOR TO G/B
```

```
@6,22 TO 20,61 DOUBLE
@10,31 TO 10,51
SET COLOR TO W/B
@8,35 SAY 'RECORD TYPE 9'
@9,32 SAY 'OUTPUT OPTIONS DATA'
@ROW()+2,24 SAY ' Echo of Input Factors: ' GET INPUT_ECHO
@ROW()+1,24 SAY ' Support Equipment Input Values: ' GET
SE_INPUT
@ROW()+1,24 SAY ' SE Cross-Reference Table: ' GET SE_CRS_REF
@ROW()+1,24 SAY ' Computed SE Costs: ' GET SE_CSTS
@ROW()+1,24 SAY ' SE Requirements: ' GET SE_RQMTS
@ROW()+1,24 SAY ' Repair Level Decisions Details: ' GET
RL_DEC_DET
@ROW()+1,24 SAY ' LRU Repair Location Summary: ' GET
LRU_RL_SUM
@ROW()+1,24 SAY ' Desired Output Units: ' GET OUTP_UNITS
READ
*WAIT
CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*EWHOLFAC.PRG - EDITS WHOLESALE EXCLUSION FACTORS DATA
SET COLOR TO W/B
CLEAR
USE WC_EX_FC
IF EOF()
APPEND BLANK
ENDIF
SET CONFIRM ON
SET COLOR TO G/B
@1,16 TO 11,69 DOUBLE
@12,16 TO 23,69 DOUBLE
@5,28 TO 5,53
@15,31 TO 15,49
SET COLOR TO W/B
@3,34 SAY 'RECORD TYPE 10'
@4,29 SAY 'WHOLESALE CHANGE FACTORS'
SET COLOR TO W/B
@ROW()+2,18 SAY ' Multiply all LRU and LRU piece part costs'
@ROW()+1,23 SAY ' and SRU and SRU piece part costs: ' GET
LRU_SRU_CS
@ROW()+1,18 SAY ' Multiply all LRU MTBFS: ' GET LRU_MTBFS
@ROW()+1,18 SAY ' Multiply all SE costs: ' GET SE_CSTS
SET COLOR TO W/B
@14,32 SAY 'EXCLUSION FACTORS'
SET COLOR TO W/B
@ROW()+2,18 SAY ' Exclude depot repair for all LRUs: ' GET
```

```
X_LRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all LRUS: ' GET
X_LRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate repair for all LRUs: '
GET X_LRU_IREP
@ROW()+1,18 SAY ' Exclude depot repair for all SRUs: ' GET
X_SRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all SRUs: ' GET
X_SRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate for all SRUs: ' GET
X_SRU_IREP
READ
*WAIT
CLOSE DATABASES
CLEAR
RETURN
Z

**********************************************************
**********************************************************

*EINDFAC.PRG -
*EDITS INDIVIDUAL CHANGE FACTORS DATA

SET TALK OFF
SET BELL OFF
SET CONFIRM ON
SET COLOR TO W/B

USE IND_CNGE
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID

DO WHILE .T.
CLEAR
STORE SPACE(1) TO CHOICE1
SET COLOR TO G/B
@4,18 TO 18,65 DOUBLE
@8,29 TO 8,55
SET COLOR TO W/B
@6,36 SAY 'RECORD TYPE 11'
@ROW()+1,30 SAY 'INDIVIDUAL CHANGE FACTORS'
@ROW()+3,35 SAY ' DD A RECORD'
@ROW()+1,34 SAY ' ELETE A RECORD'
@ROW()+1,35 SAY ' DIT A RECORD'
@ROW()+1,32 SAY ' ETURN TO EDIT MENU'
@ROW()+3,26 SAY 'ENTER             SELECTION: '

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/B
ENDIF
```

```
@10,35 SAY 'A'
@ROW()+1,34 SAY 'D'
@ROW()+1,35 SAY 'E'
@ROW()+1,32 SAY 'R'
@ROW()+3,32 SAY 'HIGHLIGHTED'
@ROW(),55 GET CHOICE1 PICTURE '@A'

READ
SET COLOR TO W/B

DO CASE
  CASE UPPER(CHOICE1) = 'R'
     CLOSE DATABASES
     RETURN

  CASE UPPER(CHOICE1) = 'A'
     SET COLOR TO W/B
     STORE 'E' TO MODULE
     DO AINDFAC

  CASE UPPER(CHOICE1) = 'D'
     CLEAR
SET COLOR TO W/B
@1,16 SAY 'RECORD TYPE 11 - INDIVIDUAL CHANGE FACTORS DATA'

SET COLOR TO G/B
@2,16 TO 2,62
@4,5 TO 4,7
@ROW(),9 TO ROW(),14
@ROW(),17 TO ROW(),28
@21,22 TO 24,59 DOUBLE

SET COLOR TO W/B
@2,5 SAY 'REC'
@ROW()+1,5 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),17 SAY 'SE RESOURCES '
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_ID, SE_RES_1, SE_RES_2

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
```

```
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B
STORE SPACE(7) TO MLRU_ID
@22,25 SAY 'ENTER LRU ID TO DELETE?'
@ROW(),49 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH LRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'
  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
ENDIF

          SET DELETED ON
          STORE .F. TO CHOICE2
          CLEAR
          SET COLOR TO G/B
          @6,20 TO 14,60 DOUBLE
          SET COLOR TO W/B
          @8,23 SAY 'Confirm Record Deletion of LRU_ID'
          @ROW()+2,38 SAY MLRU_ID PICTURE "NNNNNNN"
          @ROW()+2,32 SAY '(Y or N)?'
          @ROW(),48 GET CHOICE2 PICTURE '@! Y'
          READ

               IF CHOICE2
                DELETE
               ENDIF

     CASE UPPER(CHOICE1) = 'E'
CLEAR
SET COLOR TO W/B
@1,17 SAY 'RECORD TYPE 11 - INDIVIDUAL CHANGE FACTORS DATA'

SET COLOR TO G/B
@2,16 TO 2,62
@4,5 TO 4,7
```

```
@ROW(),9 TO ROW(),14
@ROW(),17 TO ROW(),29
@21,22 TO 24,59 DOUBLE

SET COLOR TO W/B
@2,4 SAY 'REC'
@ROW()+1,4 SAY 'NO.'
@ROW(),9 SAY 'LRU ID '
@ROW(),17 SAY 'SE RESOURCES'
GO TOP
DO WHILE .NOT. EOF()
@5,1 CLEAR TO 20,60
@5,1
DISPLAY NEXT 15 LRU_ID, SE_RES_1, SE_RES_2

STORE 'Y' TO NEXT15
@23,25 SAY 'SHOW NEXT 15 RECORDS (Y OR N)?'
@ROW(),56 GET NEXT15 PICTURE '@L'
READ

IF UPPER(NEXT15) = 'N'
  GO BOTTOM
  SKIP
ENDIF
ENDDO

@22,24 CLEAR TO 23,57
SET COLOR TO W/B

STORE SPACE(7) TO MLRU_ID
@22,26 SAY 'ENTER LRU ID TO EDIT?'
@ROW(),49 GET MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,28 SAY '(ENTER "9999" FOR NO EDIT)'
READ

IF MLRU_ID = "9999"
  CLEAR
  CLOSE DATABASES
  RETURN
ENDIF
SEEK MLRU_ID

*INVALID LRU_ID ENTERED
IF EOF()
  @22,24 CLEAR TO 23,57
  @22,34 SAY 'NO SUCH LRU ID'
  @ROW()+1,23 SAY 'PRESS ANY KEY TO RETURN TO EDIT MENU'

  ?? CHR(7)
  WAIT""
  CLOSE DATABASES
  RETURN
```

```
ENDIF

*VALID LRU_ID
STORE SE_RES_1 TO MSE_RES_1
STORE SE_CH_FAC1 TO MSE_CH_FAC1
STORE SE_RES_2 TO MSE_RES_2
STORE SE_CH_FAC2 TO MSE_CH_FAC2
STORE LRU_CST_FA TO MLRU_CST_FA
STORE MTBF_FAC TO MMTBF_FAC

CLEAR
SET COLOR TO G/B
@7,18 TO 20,63 DOUBLE
@11,27 TO 11,53
SET COLOR TO W/B
@9,34 SAY 'RECORD TYPE 11'
@ROW()+1,28 SAY 'INDIVIDUAL CHANGE FACTORS'

@ROW()+2,20 SAY ' SE Resource Numbers: ' GET MSE_RES_1
PICTURE "####"
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET MSE_CH_FAC1
PICTURE "#####.##"
@ROW()+1,20 SAY ' SE Resource Numbers: ' GET MSE_RES_2
PICTURE "####"
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET MSE_CH_FAC2
PICTURE "#####.##"
@ROW()+1,20 SAY ' LRU Identifier (work unit code): ' GET
MLRU_ID PICTURE "NNNNNNN"
@ROW()+1,20 SAY ' LRU Cost Factor: ' GET MLRU_CST_FA PICTURE
"#####.##"
@ROW()+1,20 SAY ' LRU MTBF: ' GET MMTBF_FAC PICTURE
"#####.##"
READ

REPLACE SE_RES_1 WITH MSE_RES_1
REPLACE SE_CH_FAC1 WITH MSE_CH_FAC1
REPLACE SE_RES_2 WITH MSE_RES_2
REPLACE SE_CH_FAC2 WITH MSE_CH_FAC2
REPLACE LRU_ID WITH MLRU_ID
REPLACE LRU_CST_FA WITH MLRU_CST_FA
REPLACE MTBF_FAC WITH MMTBF_FAC
ENDCASE

PACK
ENDDO
CLOSE DATABASES
SET TALK ON
SET DELETED OFF
SET COLOR TO W/B
CLEAR
RETURN
^Z
```

```
**********************************************************
**********************************************************

*AINDFAC.PRG -
*ADDS INDIVIDUAL CHANGE FACTOR DATA

DO WHILE .T.
APPEND BLANK
SKIP -1
CLEAR
SET COLOR TO G/B
@6,18 TO 19,63 DOUBLE
@10,27 TO 10,53
SET COLOR TO W/B
@8,34 SAY 'RECORD TYPE 11'
@ROW()+1,28 SAY 'INDIVIDUAL CHANGE FACTORS'
@ROW()+2,20 SAY ' SE Resource Numbers: ' GET SE_RES_1
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET SE_CH_FAC1
@ROW()+1,20 SAY ' SE Resource Numbers: ' GET SE_RES_2
@ROW()+1,20 SAY ' SE Desired Change Factor: ' GET SE_CH_FAC2
@ROW()+1,20 SAY ' LRU Identifier (work unit code): ' GET
LRU_ID
@ROW()+1,20 SAY ' LRU Cost Factor: ' GET LRU_CST_FA
@ROW()+1,20 SAY ' LRU MTBF: ' GET MTBF_FAC
READ

IF LRU_ID = "       "
*APPARANT BLANK RECORD
@20,18 CLEAR TO 22,65
STORE 'Y' TO ABORT
? CHR(7)
SET COLOR TO G/B
@20,18 TO 23,63 DOUBLE
SET COLOR TO W/B
@21,25 SAY '  Cannot Enter a Blank Record!  '
@22,21 SAY 'Abort new record addition? (Y or N)?'
@22,59 GET ABORT PICTURE '@L'
READ
 IF UPPER(ABORT) = 'Y'
   SET CONSOLE OFF
   DELETE
   PACK
   SET CONSOLE ON
 ENDIF
ENDIF

STORE 'Y' TO ANOTHER
@20,18 CLEAR TO 23,65
SET COLOR TO G/B
@20,18 TO 22,63 DOUBLE
SET COLOR TO W/B
```

```
@21,25 SAY 'Enter Another Record (Y or N)? '
@21,57 GET ANOTHER PICTURE '@L'
READ

IF MODULE = 'C'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
   PACK
   SET CONSOLE ON
     STORE 'N' TO EXIT
     @21,24 CLEAR TO 21,60
     SET COLOR TO W/B
     @21,25 SAY 'Exit Creation Module (Y or N)?'
     @21,57 GET EXIT PICTURE '@L'
     READ
       DO CASE
          CASE UPPER(EXIT) = 'Y'
          RETURN TO MASTER
          CASE UPPER(EXIT) = 'N'
          RETURN
       ENDCASE
ENDCASE
ENDIF

IF MODULE = 'E'
DO CASE
   CASE UPPER(ANOTHER) = 'N'
   SET CONSOLE OFF
    PACK
   SET CONSOLE ON
   RETURN
ENDCASE
*ENDDO
ENDIF

ENDDO

SET BELL ON
RETURN
^Z

**********************************************************
**********************************************************

*CREATE.PRG -
*CREATE MENU

SET BELL OFF
set talk off
STORE SPACE(1) TO NUMB
```

```
DO WHILE .T.
SET CONFIRM ON
SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@1,14 TO 19,65 DOUBLE
@4,17 to 4,62
@21,21 TO 23,59 DOUBLE

SET COLOR TO N/W
@3,19 say 'WHICH RECORD TYPE WOULD YOU LIKE TO CREATE?'

SET COLOR TO B/W
@5,31 SAY 'EAPONS SYSTEM DATA'
@ROW()+1,29 SAY 'AINTENANCE SYSTEM DATA'
@ROW()+1,31 SAY 'S'
@ROW(),33 SAY 'PPLY SYSTEM DATA'
@ROW()+1,29 SAY 'SUPPORT '
@ROW(),38 SAY 'QUIPMENT DATA'
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,29 SAY 'LRU '
@ROW(),34 SAY 'AILURE MOCE DATA'
@ROW()+1,37 SAY 'RU DATA'
@ROW()+1,24 SAY 'LRU/SRU/SE '
@ROW(),36 SAY 'ROSS REFERENCE DATA'
@ROW()+1,34 SAY 'UTPUT OPTIONS'
@ROW()+1,31 SAY 'W'
@ROW(),33 SAY 'OLESALE FACTORS'
@ROW()+1,32 SAY 'NDIVIDUAL FACTORS'
@ROW()+1,31 SAY 'ETURN TO MAIN MENU'
@22,25 SAY 'ENTER '
@22,43 SAY ' SELECTION: '
@22,55 GET NUMB PICTURE '@!'

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/W
ENDIF
@5,30 SAY 'W'
@ROW()+1,28 SAY 'M'
@ROW()+1,32 SAY 'U'
@ROW()+1,37 SAY 'E'
@ROW()+1,36 SAY 'L'
@ROW()+1,33 SAY 'F'
@ROW()+1,36 SAY 'S'
@ROW()+1,35 SAY 'C'
@ROW()+1,33 SAY 'O'
@ROW()+1,32 SAY 'H'
@ROW()+1,31 SAY 'I'
@ROW()+1,30 SAY 'R'
@22,31 SAY 'HIGH-LIGHTED'
```

```
READ
do case
 CASE NUMB = 'R'
  RETURN TO MASTER
endcase
do warn
do case
 case numb = 'W'
  do CWEAPSYS
 case numb = 'M'
  do CMAINTSYS
 case numb = 'U'
  do CSUPSYS
 case numb = 'E'
  do CSESYS
 case numb = 'L'
  do CLRU
 CASE NUMB = 'F'
  DO CLRUFM
 CASE NUMB = 'S'
  DO CSRU
 CASE NUMB = 'C'
  DO CCROSS
 CASE NUMB = 'O'
  DO COUTOPT
 CASE NUMB = 'H'
  DO CWHOLFAC
 CASE NUMB = 'I'
  DO CINDFAC
 ENDCASE
ENDDO
^Z

**********************************************************
**********************************************************

*WARN.PRG -
SET BELL OFF
set talk off
STORE SPACE(1) TO GOON

DO WHILE .T.
SET CONFIRM ON
SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@6,14 TO 19,65 DOUBLE
@21,21 TO 23,59 DOUBLE

SET COLOR TO B/W
@8,19 say 'THIS MODULE WILL ERASE THE DATA CONTAINED'
@ROW()+2,19 SAY 'IN THE DATABASE FILE FOR THIS RECORD TYPE.'
```

```
@ROW()+2,19 SAY 'THE CURRENTLY STORED DATA WILL BE REPLACED'
@ROW()+2,19 SAY 'BY THE DATA ENTERED FOLLOWING THIS PROMPT'
@ROW()+2,19 SAY 'DO YOU WANT TO CONTINUE OR RETURN TO THE'
@ROW()+2,35 SAY 'MAIN MENU?'
@22,25 SAY 'ENTER HIGH-LIGHTED SELECTION: '
@ROW(),55 GET GOON PICTURE '@!'

SET COLOR TO N/W
IF ISCOLOR()
SET COLOR TO R/W
ENDIF
@16,34 SAY 'C'
@ROW(),46 SAY 'R'
@22,31 SAY 'HIGH-LIGHTED'


READ
do case
 case GOON = 'C'
  RETURN
 CASE GOON = 'R'
  RETURN TO MASTER
 ENDCASE
ENDDO
^Z

**********************************************************
**********************************************************

*CWEAPSYS.PRG - CREATES WEAPONS SYSTEM DATA
SET COLOR TO W/B
CLEAR
USE WEAPON_SY
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
SET CONFIRM ON
SET COLOR TO G/B
@2,8 TO 20,70 DOUBLE
@6,30 TO 6,49
SET COLOR TO W/B
@4,33 SAY 'RECORD TYPE 1'
@ROW()+1,31 SAY 'WEAPON SYSTEM DATA'
SET COLOR TO W/B
@ROW()+2,10 SAY ' End Item Name: ' GET END_ITEM
@ROW()+1,10 SAY ' No. of Intermediate Locations: ' GET
NO_I_LOC
@ROW()+1,10 SAY ' Ratio Force Overseas: ' GET RATIO_OS
@ROW()+1,10 SAY ' Years System Life: ' GET YRS_SYS_LI
@ROW()+1,10 SAY ' Equivalent Weapon Systems Per Intermediate
Location: ' GET EQ_WS_ILOC
```

```
@ROW()+1,10 SAY ' System Operation Hours Per Month: ' GET
MO_OPS_HRS
@ROW()+1,10 SAY ' SE Development Cost: ' GET SE_DEV_CST

@ROW()+1,10 SAY ' Sensitivity Alternatives: ' GET SENSI_ALT

@ROW()+1,10 SAY ' Lower Sensitivity Range: ' GET L_SEN_RANGE

@ROW()+1,10 SAY ' Upper Sensitivity Range: ' GET U_SEN_RANGE

@ROW()+1,10 SAY ' Sensitivity Type: ' GET SENSI_TYPE
@ROW()+1,10 SAY ' Optional Information: ' GET OPTION_INF

READ
SET COLOR TO G/B
@21,8 TO 23,70 DOUBLE
SET COLOR TO W/B
STORE 'N' TO EXIT
@22,25 SAY 'Exit Creation Module (Y or N)?'
@22,57 GET EXIT PICTURE '@L'
READ
  DO CASE
     CASE UPPER(EXIT) = 'Y'
     RETURN TO MASTER
     CASE UPPER(EXIT) = 'N'
     RETURN
  ENDCASE

?
?
?
*WAIT
SET CONFIRM OFF
CLOSE DATABASES
SET COLOR TO W/B
CLEAR

RETURN
^Z

**********************************************************
**********************************************************

*CMAINTSY.PRG - CREATES MAINTENANCE SYSTEM DATA - 26-4-88
SET COLOR TO W/B
SET CONFIRM ON
CLEAR
USE MAINT_SYS
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
```

```
SET COLOR TO G/B
@7,22 TO 19,59 DOUBLE
@11,28 TO 11,52
SET COLOR TO W/B
@9,34 SAY 'RECORD TYPE 2'
@ROW()+1,29 SAY 'MAINTENANCE SYSTEM DATA'
@ROW()+2,24 SAY ' Intermediate Shop Man-hours: ' GET
INT_MNHRS
@ROW()+1,24 SAY ' Intermediate Labor Rate: ' GET INT_LAB_RT
@ROW()+1,24 SAY ' Depot Shop Man-hours: ' GET DEP_MNHRS
@ROW()+1,24 SAY ' Depot Labor Rate: ' GET DEP_LAB_RT
@ROW()+1,24 SAY ' Intermediate Turnover Rate: ' GET
INT_TRN_RT
@ROW()+1,24 SAY ' Depot Turnover Rate: ' GET DEP_TRN_RT
READ

SET COLOR TO G/B
@21,22 TO 23,59 DOUBLE
SET COLOR TO W/B
STORE 'N' TO EXIT
@22,25 SAY 'Exit Creation Module (Y or N)?'
@22,57 GET EXIT PICTURE '@L'
READ
  DO CASE
     CASE UPPER(EXIT) = 'Y'
     RETURN TO MASTER
     CASE UPPER(EXIT) = 'N'
     RETURN
  ENDCASE

*WAIT
CLOSE DATABASES
SET CONFIRM OFF
CLEAR
RETURN
Z

**********************************************************
**********************************************************

*CSUPSYS.PRG -
*CREATES SUPPLY SYSTEM DATA
SET COLOR TO W/B
CLEAR
USE SUPPLY_SYS
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
SET CONFIRM ON
SET COLOR TO G/B
@3,21 TO 21,62 DOUBLE
```

```
@7,32 TO 7,51
SET COLOR TO W/B
@5,36 SAY 'RECORD TYPE 3'
@ROW()+1,33 SAY 'SUPPLY SYSTEM DATA'
@ROW()+2,23 SAY ' Initial Management Cost: ' GET INI_MGT_CS

@ROW()+1,23 SAY ' Recurring Management Cost: ' GET REC_MGT_CS

@ROW()+1,23 SAY ' Base Supply Management Cost: ' GET
BS_MGT_CS
@ROW()+1,23 SAY ' Order and Ship Time CONUS: ' GET CONUS_SHIP

@ROW()+1,23 SAY ' Order and Ship Time Overseas: ' GET OS_SHIP

@ROW()+1,23 SAY ' Packing Cost CONUS: ' GET CONUS_PK_C
@ROW()+1,23 SAY ' Packing Cost Overseas: ' GET OS_PK_CST
@ROW()+1,23 SAY ' Packed Wt Ratio CONUS: ' GET CONUS_PKWT
@ROW()+1,23 SAY ' Packed Wt Ratio Overseas: ' GET OS_PKWT_RT

@ROW()+1,23 SAY ' Shipping Rate CONUS: ' GET CONUS_SHRT
@ROW()+1,23 SAY ' Shipping Rate Overseas: ' GET OS_SHIP_RT
@ROW()+1,23 SAY ' Tech Data Cost: ' GET TEC_DAT_CS
READ
SET COLOR TO G/B
@22,21 TO 24,62 DOUBLE
SET COLOR TO W/B
STORE 'N' TO EXIT
@23,25 SAY 'Exit Creation Module (Y or N)?'
@23,57 GET EXIT PICTURE '@L'

READ
  DO CASE
     CASE UPPER(EXIT) = 'Y'
     RETURN TO MASTER
     CASE UPPER(EXIT) = 'N'
     RETURN
  ENDCASE

*WAIT
CLOSE DATABASES
SET CONFIRM OFF
CLEAR
RETURN
Z

**********************************************************
**********************************************************

*CSESYS.PRG -
*CREATES SUPPORT EQUIPMENT DATA
SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE SE_DATA
ERASE ISERN.NDX
INDEX ON SERN TO ISERN
APPEND BLANK
DELETE ALL
PACK
*APPEND BLANK
SET CONFIRM ON

STORE 'C' TO MODULE
DO ASESYS

SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z

******************************************************************
******************************************************************

*CLRU.PRG -
*CREATES LRU DATA
SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE LRU_DATA
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID
APPEND BLANK
DELETE ALL
PACK
SET CONFIRM ON

STORE 'C' TO MODULE
DO ALRU
*ERASE SE_DATA.DBS
*SORT ON LRU_ID TO LRU_DATA.DBS

SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z

******************************************************************
******************************************************************
```

```
*CLRUFM.PRG -
*CREATES LRU FAILURE MODE DATA

SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE LRU_FM_D
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID
APPEND BLANK
DELETE ALL
PACK
SET CONFIRM ON

STORE 'C' TO MODULE
DO ALRUFM

SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z^Z^Z
**********************************************************
**********************************************************

*CSRU.PRG -
*CREATES SRU DATA
SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE SRU_DATA
ERASE ISID.NDX
INDEX ON SRU_ID TO ISID
APPEND BLANK
DELETE ALL
PACK
SET CONFIRM ON
STORE 'C' TO MODULE
DO ASRU
SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************
```

```
*CCROSS.PRG -
*CREATES LRU/SRU/SE CROSS REFERENCE DATA
SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE CROSS_RE
ERASE ILSID.NDX
INDEX ON LRU_SRU_ID TO ILSID
APPEND BLANK
DELETE ALL
PACK
SET CONFIRM ON
STORE 'C' TO MODULE
DO ACROSS
SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*COUTOPT.PRG - CREATES OUTPUT OPTIONS DATA - April 26, 1988
SET COLOR TO W/B
SET CONFIRM ON
CLEAR
USE OUTP_OPT
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
SET COLOR TO G/B
@6,22 TO 20,61 DOUBLE
@10,31 TO 10,51
SET COLOR TO W/B
@8,35 SAY 'RECORD TYPE 9'
@9,32 SAY 'OUTPUT OPTIONS DATA'
@ROW()+2,24 SAY ' Echo of Input Factors: ' GET INPUT_ECHO
@ROW()+1,24 SAY ' Support Equipment Input Values: ' GET
SE_INPUT
@ROW()+1,24 SAY ' SE Cross-Reference Table: ' GET SE_CRS_REF

@ROW()+1,24 SAY ' Computed SE Costs: ' GET SE_CSTS
@ROW()+1,24 SAY ' SE Requirements: ' GET SE_RQMTS
@ROW()+1,24 SAY ' Repair Level Decisions Details: ' GET
RL_DEC_DET
@ROW()+1,24 SAY ' LRU Repair Location Summary: ' GET
LRU_RL_SUM
```

```
@ROW()+1,24 SAY ' Desired Output Units: ''GET OUTP_UNITS
READ

SET COLOR TO G/B
@21,22 TO 23,61 DOUBLE
SET COLOR TO W/B
STORE 'N' TO EXIT
@22,25 SAY 'Exit Creation Module (Y or N)?'
@22,57 GET EXIT PICTURE '@L'
READ
  DO CASE
     CASE UPPER(EXIT) = 'Y'
     RETURN TO MASTER
     CASE UPPER(EXIT) = 'N'
     RETURN
  ENDCASE

*WAIT
CLOSE DATABASES
SET CONFIRM OFF
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*CWHOLFAC.PRG - CREATES WHOLESALE EXCLUSION FACTORS DATA
SET COLOR TO W/B
CLEAR
USE WC_EX_FC
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
SET CONFIRM ON
SET COLOR TO G/B
@1,16 TO 10,69 DOUBLE
@11,16 TO 21,69 DOUBLE
@4,29 TO 4,54
@14,32 TO 14,50
SET COLOR TO W/B
@2,35 SAY 'RECORD TYPE 10'
@3,30 SAY 'WHOLESALE CHANGE FACTORS'
SET COLOR TO W/B
@ROW()+2,18 SAY ' Multiply all LRU and LRU piece part costs'
@ROW()+1,23 SAY ' and SRU and SRU piece part costs: ' GET
LRU_SRU_CS
@ROW()+1,18 SAY ' Multiply all LRU MTBFS: ' GET LRU_MTBFS
@ROW()+1,18 SAY ' Multiply all SE costs: ' GET SE_CSTS
SET COLOR TO W/B
@12,33 SAY 'EXCLUSION FACTORS'
```

```
SET COLOR TO W/B
@ROW()+2,18 SAY ' Exclude depot repair for all LRUs: ' GET
X_LRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all LRUS: ' GET
X_LRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate repair for all LRUs: '
GET X_LRU_IREP
@ROW()+1,18 SAY ' Exclude depot repair for all SRUs: ' GET
X_SRU_DREP
@ROW()+1,18 SAY ' Exclude scrap for all SRUs: ' GET
X_SRU_SCRP
@ROW()+1,18 SAY ' Exclude intermediate for all SRUs: ' GET
X_SRU_IREP
READ

SET COLOR TO G/B
@22,16 TO 24,69 DOUBLE
SET COLOR TO W/B
STORE 'N' TO EXIT
@23,25 SAY 'Exit Creation Module (Y or N)?'
@23,57 GET EXIT PICTURE '@L'
READ
  DO CASE
     CASE UPPER(EXIT) = 'Y'
     RETURN TO MASTER
     CASE UPPER(EXIT) = 'N'
     RETURN
  ENDCASE

CLOSE DATABASES
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*CINDFAC.PRG -
*CREATES INDIVIDUAL CHANGE FACTOR DATA
SET TALK OFF
SET BELL OFF
SET COLOR TO W/B
CLEAR
USE IND_CNGE
ERASE ILID.NDX
INDEX ON LRU_ID TO ILID
APPEND BLANK
DELETE ALL
PACK
SET CONFIRM ON

STORE 'C' TO MODULE
```

```
DO AINDFAC.PRG

SET CONSOLE ON
CLOSE DATABASES
SET COLOR TO W/B
CLEAR
RETURN
^Z

**********************************************************
**********************************************************

*Run.prg
SET CONFIRM ON
SET COLOR TO B/W

SET BELL OFF

clear
set talk off
STORE SPACE(1) TO CHOICE
DO WHILE .T.
SET COLOR TO B/W
CLEAR
SET COLOR TO G/W
@4,20 TO 18,58 DOUBLE
@19,21 TO 21,57 DOUBLE
@8,24 TO 8,54


SET COLOR TO N/W
@6,25 say 'NETWORK REPAIR LEVEL ANALYSIS'

@ROW()+1,32 SAY 'MODEL RUN MENU'
SET COLOR TO B/W
@ROW()+2,32 SAY 'RUN '
@ROW(),37 SAY 'RLA MODEL '
@ROW()+2,28 SAY 'RUN  PS '
@ROW(),36 SAY 'MAX COST MODEL '
@ROW()+2,30 SAY 'SHOW DIRECTORY OF'

@ROW()+1,27 SAY '.DAT FILES ON HARD DRIVE'

@ROW()+2,31 SAY 'ETURN TO MAIN MENU '
@20,24 SAY 'ENTER '
@ROW(),42 SAY ' SELECTION: '
@ROW(),54 GET CHOICE PICTURE '@!'

SET COLOR TO N/W

IF ISCOLOR()
SET COLOR TO R/W
```

```
ENDIF
@9,36 SAY 'N'
@ROW()+2,32 SAY 'T'
@ROW()+2,35 SAY 'D'

@ROW()+3,30 SAY 'R'
@20,30 SAY 'HIGH-LIGHTED'

READ
do case
  case choice = 'R'
   RETURN
  CLEAR
  case choice = 'N'

   DO DATE
   DO SALL
   RUN NRLA
   SET CONFIRM ON
 RETURN
  case choice = 'T'
    RUN B1BTPS
    SET CONFIRM ON
    RETURN
  CASE CHOICE = 'D'
    SET COLOR TO B/W
    CLEAR
    DIR *.DAT
    WAIT
clear
set talk on
endcase
RETURN
ENDDO
^Z

**********************************************************
**********************************************************

*DATE.PRG -
*SAVES TODAY'S DATE
SET CONFIRM ON
SET COLOR TO B/W
SET BELL OFF
clear
set talk off

USE DATE.DBF
APPEND BLANK
DELETE ALL
PACK
APPEND BLANK
```

```
CLEAR
SET COLOR TO G/W
@7,20 TO 20,58 DOUBLE
@11,29 TO 11,49

SET COLOR TO B/W
@10,25 say "ENTER TODAY'S DATE (MM/DD/YR)"
@ROW()+3,36 SAY 'MONTH: ' GET MONTH
@ROW()+2,36 SAY 'DAY: ' GET DAY
@ROW()+2,36 SAY 'YEAR: ' GET YEAR
READ
SET CONFIRM OFF
CLOSE DATABASES
clear
set talk on
^Z

**********************************************************
**********************************************************

*SALL.PRG - SAVES ALL DATA
CLEAR
SET COLOR TO B/W
CLOSE ALL
SET CONFIRM ON
SET COLOR TO G/W
@4,7 TO 21,75 DOUBLE

SET COLOR TO B/W
STORE SPACE(12) TO RUNSPEC
STORE SPACE(12) TO PRGSPEC
@7,27 SAY 'ENTER FILENAMES TO STORE DATA'
@9,15 SAY 'FILENAMES MUST BE ALPHA-NUMERICS UP TO 8
CHARACTERS'
@10,30 SAY 'WITH A .DAT EXTENSION.'
@16,10 SAY 'ENTER FILENAME FOR RUN SPECIFIC DATA:'
SET COLOR TO N/W
@10,37 SAY '.DAT'
@16,49 GET RUNSPEC PICTURE "XXXXXXXXXXXX"
SET COLOR TO B/W
@18,10 SAY 'ENTER FILENAME FOR PROGRAM SPECIFIC (NRLA) DATA:'

SET COLOR TO N/W
@ROW(),61 GET PRGSPEC PICTURE "XXXXXXXXXXXX"
READ

SET COLOR TO B/W
CLEAR
SET CONSOLE ON
SET COLOR TO G/W
@4,12 TO 15,69 DOUBLE
SET COLOR TO B/W
```

```
@6,22 SAY 'RUN SPECIFIC DATA IS BEING WRITTEN TO:'
SET COLOR TO N/W
IF ISCOLOR()
 SET COLOR TO R/W
ENDIF
@ROW()+2,34 SAY RUNSPEC
SET CONSOLE OFF

SET ALTERNATE TO &RUNSPEC
SET ALTERNATE ON
DO SOUTOPT
DO SWHOLFAC
DO SINDFAC
SET ALTERNATE OFF
CLOSE ALTERNATE
CLOSE ALL

SET COLOR TO B/W
@11,17 SAY 'PROGRAM SPECIFIC (NRLA) DATA IS BEING WRITTEN
TO:'
SET COLOR TO N/W
IF ISCOLOR()
 SET COLOR TO R/W
ENDIF
@ROW()+2,34 SAY PRGSPEC
SET COLOR TO B/W
@16,1
SET CONSOLE OFF

SET ALTERNATE TO &PRGSPEC
SET ALTERNATE ON
DO SWEAPSYS
DO SMAINTSYS
DO SSUPSYS
DO SSESYS
DO SLRU
DO SLRUFM
DO SSRU
DO SCROSS
CLOSE ALTERNATE
SET ALTERNATE OFF
CLOSE ALL
SET CONSOLE ON

^Z

**********************************************************
**********************************************************

*SWEAPSYS.PRG -
*SAVES WEAPONS SYSTEM DATA IN NRLA FORMAT
SET TALK OFF
```

```
USE WEAPON_S.DBF
RO = RATIO_OS * 100
LSR = L_SEN_RANGE * 100
USR = U_SEN_RANGE * 100
STORE TRANSFORM(RO,"999") TO RO
STORE TRANSFORM(LSR,"99") TO LSR
STORE TRANSFORM(USR,"999") TO USR
??END_ITEM + '  ';
+ STR(NO_I_LOC,3,0) + ' ';
+ RO + ' ';
+ STR(YRS_SYS_LI,2,0);
+ STR(EQ_WS_ILOC,4,0) + ' ';
+ STR(MO_OPS_HRS,3,0);
+ STR(SE_DEV_CST,7,0);
+ STR(SENSI_ALT,1,0) + '      ';
+LSR + ' ';
+USR + ' '
??STR(SENSI_TYPE,1,0) + OPTION_INF
CLOSE DATABASES
RETURN
^Z
*+ STR(L_SEN_RANGE,2,0) + ' ';
*+ STR(U_SEN_RANGE,3,0) + ' '
^Z

*********************************************************
*********************************************************

*SMAINTSY.PRG -
* SAVES MAINTENANCE SYSTEM DATA IN NRLA FORMAT
SET TALK OFF

USE MAINT_SYS
ILR = INT_LAB_RT * 100
DLR = DEP_LAB_RT * 100
STORE TRANSFORM(ILR,"9999") TO ILR
STORE TRANSFORM(DLR,"9999") TO DLR
?STR(INT_MNHRS,3,0) + ' ';
+ ILR + ' ';
+ STR(DEP_MNHRS,3,0) + ' ';
+ DLR + ' ';
+ STR(INT_TRN_RT,4,3) + ' ';
+ STR(DEP_TRN_RT,4,3)
CLOSE DATABASES
RETURN
^Z

*********************************************************
*********************************************************
```

```
*SSUPSYS.PRG -
*SAVES SUPPLY SYSTEM DATA IN NRLA FORMAT
SET TALK OFF
SET DECIMALS TO 3
USE SUPPLY_S.DBF

*CHECK LENGTH OF INPUT VARIABLES FOR READ FIELDS WITH
FLOATING DECIMAL POINTS

STORE ' ' TO VAR4
STORE INI_MGT_CS TO VAR1
STORE TRANSFORM(INI_MGT_CS,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO IMC

STORE REC_MGT_CS TO VAR1
STORE TRANSFORM(REC_MGT_CS,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO RMC

STORE BS_MGT_CS TO VAR1
STORE TRANSFORM(BS_MGT_CS,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO BMC

STORE CONUS_PK_C TO VAR1
STORE TRANSFORM(CONUS_PK_C,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO CPC

STORE OS_PK_CST TO VAR1
STORE TRANSFORM(OS_PK_CST,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO OPC

STORE CONUS_PKWT TO VAR1
STORE TRANSFORM(CONUS_PKWT,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO CPW

STORE OS_PKWT_RT TO VAR1
STORE TRANSFORM(OS_PKWT_RT,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO OPR

STORE CONUS_SHRT TO VAR1
STORE TRANSFORM(CONUS_SHRT,"999.999") TO VAR2
DO SUPCKL4
STORE VAR4 TO CSR

STORE OS_SHIP_RT TO VAR1
STORE TRANSFORM(OS_SHIP_RT,"999.999") TO VAR2
```

```
DO SUPCKL4
STORE VAR4 TO OSR

STORE TEC_DAT_CS TO VAR1
STORE TRANSFORM(TEC_DAT_CS,"999.99") TO VAR2
DO SUPCKL41
STORE VAR4 TO TDC

?IMC + ' ';
+ RMC + ' ';
+ BMC + ' ';
+ STR(CONUS_SHIP,4,3) + ' ';
+ STR(OS_SHIP,4,3) + ' ';
+ CPC + ' ';
+ OPC + ' ';
+ CPW + ' ';
+ OPR + ' '
??CSR + ' ';
+ OSR + ' ';
+ TDC
CLOSE DATABASES
SET TALK ON
RETURN
^Z

**********************************************************
**********************************************************

*CKLNGTH5.PRG -
*CHECKS LENGTH OF INPUT VARIABLES FOR READ FIELDS OF FIVE
COLUMNS WITH FLOATING DECIMAL POINTS AND TRANSFORMS INTO
PROPER FORMAT
SET TALK OFF

STORE (LTRIM(VAR2)) TO VAR3

IF LEN(LTRIM(VAR3)) > 6
     STORE TRANSFORM (VAR1,"9999.") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) = 6
     STORE TRANSFORM (VAR1, "999.9") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) <= 5
     STORE TRANSFORM (VAR1, "99.99") TO VAR4
     RETURN
ENDIF
^Z
```

```
**********************************************************
**********************************************************

*SUPCKL4.PRG -
*CHECKS LENGTH OF SUPPLY SYSTEM INPUT VARIABLES FOR READ
FIELDS OF FOUR COLUMNS WITH FLOATING DECIMAL POINTS AND
TRANSFORMS INTO PROPER FORMAT
SET TALK OFF

STORE (LTRIM(VAR2)) TO VAR3

IF LEN(LTRIM(VAR3)) > 6
     STORE TRANSFORM (VAR1,"999.") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) = 6
          STORE TRANSFORM (VAR1, "99.9") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) <= 5
     IF INT(VAR1) = 0
          STORE SUBSTR(VAR3,2,5) TO VAR4
     ELSE
          STORE TRANSFORM (VAR1, "9.99") TO VAR4
     RETURN
ENDIF
^Z

**********************************************************
**********************************************************

*SUPCKL41.PRG -
*CHECKS LENGTH OF SUPPLY SYSTEM INPUT VARIABLES FOR READ
FIELDS OF FOUR COLUMNS WITH FLOATING DECIMAL POINTS AND
TRANSFORMS INTO PROPER FORMAT
SET TALK OFF

STORE (LTRIM(VAR2)) TO VAR3

IF LEN(LTRIM(VAR3)) > 5
     STORE TRANSFORM (VAR1,"999.") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) = 5
          STORE TRANSFORM (VAR1, "99.9") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) <= 4
```

```
          STORE TRANSFORM (VAR1, "9.99") TO VAR4
     RETURN
ENDIF
^Z
**********************************************************
**********************************************************

*SSESYS.PRG -
*SAVES SUPPORT EQUIPMENT DATA IN NRLA FORMAT

SET TALK OFF
ERASE SESORT.DBF
USE SE_DATA.DBF
SORT ON SERN TO SESORT.DBF
USE SESORT.DBF
DO WHILE .NOT. EOF()
?'1' + STR(SERN,4,0) + ' ';
+ SE_NAME + ' ';
+ STR(SE_COST,8,2);
+ STR(SE_O_M_CST,5,2);
+ STR(NO_OF_SE,2,0) + ' ';
+ STR(CURR_USAGE,3,2) + ' ';
+ STR(AVAIL_TIME,3,2) + ' ';
+ STR(FACIL_COST,8,2)
SKIP
ENDDO
?'99'
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

*SLRU.PRG -
*SAVES LINE REPLACEABLE UNIT (LRU) DATA IN NRLA FORMAT

USE LRU_DATA.DBF
SET TALK OFF
SET DECIMALS TO 3

DO WHILE .NOT. EOF()
STORE ' ' TO VAR4

STORE OPS_RATIO TO VAR1
STORE TRANSFORM(OPS_RATIO,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO OR

STORE DEP_RCT_CO TO VAR1
STORE TRANSFORM(DEP_RCT_CO,"999.999") TO VAR2
DO CKLNGTH4
```

```
STORE VAR4 TO DRC

STORE DEP_RCT_OS TO VAR1
STORE TRANSFORM(DEP_RCT_OS,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO DRO

STORE INT_RCT_CO TO VAR1
STORE TRANSFORM(INT_RCT_CO,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO IRC

STORE R_FAIL_RIP TO VAR1
STORE TRANSFORM(R_FAIL_RIP,"99.99") TO VAR2
DO CKLNGTH3
STORE VAR4 TO RFR

?'31' + ' ' + LRU_ID + LRU_NAME;
+ STR(NO_LRU_END,3,2);
+ STR(UNIT_COST,7,2);
+ STR(WEIGHT,4,1);
+ OR;
+ DRC;
+ DRO;
+ IRC;
+ RFR;
+ STR(NO_GPSE_RQ,3,0);
+ STR(MTBF,6,2)
SKIP
ENDDO
?'99'
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

*CKLNGTH3.PRG -
*CHECKS LENGTH OF INPUT VARIABLES FOR READ FIELDS OF THREE
COLUMNS WITH FLOATING DECIMAL POINTS AND TRANSFORMS INTO
PROPER FORMAT
SET TALK OFF

STORE (LTRIM(VAR2)) TO VAR3

IF LEN(LTRIM(VAR3)) > 4
     STORE TRANSFORM (VAR1,"99.") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) = 4
```

```
     IF INT(VAR1) = 0
          STORE SUBSTR(VAR3,2,4) TO VAR4
     ELSE
          STORE TRANSFORM (VAR1, "9.9") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) <= 3
     IF INT(VAR1) = 0
          STORE SUBSTR(VAR3,2,4) TO VAR4
     ELSE
     STORE TRANSFORM (VAR1, ".99") TO VAR4
     RETURN
ENDIF
^Z

**********************************************************
**********************************************************

*SLRUFM.PRG -
* SAVES LRU FAILURE MODE DATA IN NRLA FORMAT


USE LRU_FM_D.DBF
SET TALK OFF

DO WHILE .NOT. EOF()
STORE ' ' TO VAR4

STORE FM_RATIO TO VAR1
STORE TRANSFORM(FM_RATIO,"9.999") TO VAR2
     IF INT(VAR1) = 0
          STORE SUBSTR(VAR2,3,5) TO FMR
     ELSE
          STORE TRANSFORM(VAR1,"9.9") TO FMR
     ENDIF

STORE DEP_MX_MHR TO VAR1
STORE TRANSFORM(DEP_MX_MHR,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO DMM

STORE INT_MX_MHR TO VAR1
STORE TRANSFORM(INT_MX_MHR,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO IMM

STORE WKS_MX_TRN TO VAR1
STORE TRANSFORM(WKS_MX_TRN,"999.999") TO VAR2
```

```
DO CKLNGTH4
STORE VAR4 TO WMT

STORE SE_HRS_REP TO VAR1
STORE TRANSFORM(SE_HRS_REP,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO SHR

?'41' + LRU_ID;
+ STR(FM_IDEN_NO,2,0);
+ FMR;
+ SRU_ID + FM_SRU_NAM;
+ STR(NO_NEW_PRT,3,0);
+ STR(NO_STD_PRT,3,0);
+ STR(REP_PRT_CS,5,2);
+ STR(WT_PC_PART,4,2)
??STR(NO_TRN_DEP,3,0);
+ STR(NO_TRN_INT,3,0);
+ DMM;
+ IMM;
+ WMT;
+ STR(TRNG_CST,5,2);
+ STR(TEC_DT_PGS,3,0);
+ STR(NO_SPSE_RQ,3,0);
+ STR(FRC_LRU_FM,3,0);
+ SHR
SKIP
ENDDO
?'99'
CLOSE DATABASES
RETURN
^Z



***********************************************************
***********************************************************

*CKLNGTH4.PRG -
*CHECKS LENGTH OF INPUT VARIABLES FOR READ FIELDS OF FOUR
COLUMNS WITH FLOATING DECIMAL POINTS AND TRANSFORMS INTO
PROPER FORMAT
SET TALK OFF

STORE (LTRIM(VAR2)) TO VAR3
IF LEN(LTRIM(VAR3)) > 6
     STORE TRANSFORM (VAR1,"999.") TO VAR4
     RETURN
ENDIF
```

```
IF LEN(LTRIM(VAR3)) = 6
     STORE TRANSFORM (VAR1,"99.9") TO VAR4
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) = 5
     IF INT(VAR1) = 0
          STORE SUBSTR(VAR3,2,5) TO VAR4
     ELSE
          STORE TRANSFORM (VAR1, "9.99") TO VAR4
     ENDIF
     RETURN
ENDIF

IF LEN(LTRIM(VAR3)) <= 4
     IF INT(VAR1) = 0
          STORE SUBSTR (VAR3,2,5) TO VAR4
     ELSE
          STORE TRANSFORM (VAR1, "9.99") TO VAR4
     ENDIF
     RETURN
ENDIF
^Z

**********************************************************
**********************************************************

*SSRU.PRG -
*SAVES SRU DATA IN NRLA FORMAT

USE SRU_DATA
SET TALK OFF

DO WHILE .NOT. EOF()
STORE ' ' TO VAR4

STORE DEP_RCT_CO TO VAR1
STORE TRANSFORM(DEP_RCT_CO,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO DRC

STORE DEP_RCT_OS TO VAR1
STORE TRANSFORM(DEP_RCT_OS,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO DRO

STORE INT_RCT TO VAR1
STORE TRANSFORM(INT_RCT,"999.999") TO VAR2
DO CKLNGTH4
STORE VAR4 TO IR

STORE DEP_MX_MHR TO VAR1
STORE TRANSFORM(DEP_MX_MHR,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO DMM

STORE INT_MX_MHR TO VAR1
STORE TRANSFORM(INT_MX_MHR,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO IMM

STORE MX_TRN_WKS TO VAR1
STORE TRANSFORM(MX_TRN_WKS,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO MTW

STORE SE_HRS_REP TO VAR1
STORE TRANSFORM(SE_HRS_REP,"999.999") TO VAR2

DO CKLNGTH4
STORE VAR4 TO SHR

?'51' + SRU_ID;
 + STR(SRU_CST,5,2);
 + STR(SRU_WEIGHT,4,1);
 + STR(CST_PP_REP,5,2);
 + STR(WT_PP_ASSY,4,2);
 + STR(NO_NEW_PP,3,0);
 + STR(NO_STD_PP,3,0);
 + STR(NO_PGS_T_D,3,1);
 + STR(NO_KIND_SE,2,0);
 + DRC;
 + DRO
??IR;
 + DMM;
 + IMM;
 + STR(DEP_TRNEES,3,0);
 + STR(INT_TRNEES,3,0);
 + MTW;
 + STR(MX_TRN_CST,5,2);
 + STR(FRC_SRU_DE,3,0);
 + SHR
SKIP
ENDDO
?'99'
CLOSE DATABASES
RETURN
^Z

************************************************************
************************************************************
```

```
*SCROSS.PRG -
*SAVES LRU/SRU/SE CROSS REFERENCE DATA IN NRLA FORMAT

USE CROSS_RE

SET TALK OFF
DO WHILE .NOT. EOF()
?STR(REC_TYP_ID,1,0) + '2' + ' ';
+ LRU_SRU_ID + STR(FM_ID_NO,2,0);
+ STR(SE_RES_NO1,4,0);
+ STR(SE_RES_NO2,4,0);
+ STR(SE_RES_NO3,4,0)
??STR(SE_RES_NO4,4,0);
+ STR(SE_RES_NO5,4,0);
+ STR(SE_RES_NO6,4,0);
+ STR(SE_RES_NO7,4,0);
+ STR(SE_RES_NO8,4,0);
+ STR(SE_RES_NO9,4,0)
??STR(SE_RES_N10,4,0);
+ STR(SE_RES_N11,4,0);
+ STR(SE_RES_N12,4,0);
+ STR(SE_RES_N13,4,0);
+ STR(SE_RES_N14,4,0);
+ STR(SE_RES_N15,4,0);
+ STR(SE_RES_N16,4,0);
+ STR(CONTIN_IND,2,0)
SKIP
ENDDO
?'999999'
?
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

*SOUTOPT.PRG -
*SAVES OUTPUT OPTIONS DATA IN NRLA FORMAT
USE DATE
??STR(YEAR,2,0);
+ STR(MONTH,2,0);
+ STR(DAY,2,0)
USE OUTP_OPT

?STR(INPUT_ECHO,1,0);
+ STR(SE_INPUT,1,0);
+ STR(SE_CRS_REF,1,0);
+ STR(SE_CSTS,1,0);
+ STR(SE_RQMTS,1,0);
+ STR(RL_DEC_DET,1,0);
+ STR(LRU_RL_SUM,1,0) + '   ';
+ STR(OUTP_UNITS,1,0)
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

*SWHOLFAC.PRG -
*SAVES WHOLESALE EXCLUSION FACTORS DATA IN NRLA FORMAT

USE WC_EX_FC

SET TALK OFF
STORE ' ' TO VAR4

STORE LRU_SRU_CS TO VAR1
STORE TRANSFORM(LRU_SRU_CS,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO LSC

STORE LRU_MTBFS TO VAR1
STORE TRANSFORM(LRU_MTBFS,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO LM

STORE SE_CSTS TO VAR1
STORE TRANSFORM(SE_CSTS,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO SEC

?LSC + ' ';
+ LM + ' ';
+ SEC + ' ';
+ STR(X_LRU_DREP,1,0);
+ STR(X_LRU_SCRP,1,0);
+ STR(X_LRU_IREP,1,0);
+ STR(X_SRU_DREP,1,0);
+ STR(X_SRU_SCRP,1,0);
+ STR(X_SRU_IREP,1,0) + '     '
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

*SINDFAC.PRG -
*SAVES INDIVIDUAL CHANGE FACTORS IN NRLA FORMAT
USE IND_CNGE
```

```
SET TALK OFF
STORE ' ' TO VAR4

DO WHILE .NOT. EOF()

STORE SE_CH_FAC1 TO VAR1
STORE TRANSFORM(SE_CH_FAC1,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO SCF1

STORE SE_CH_FAC2 TO VAR1
STORE TRANSFORM(SE_CH_FAC2,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO SCF2

STORE LRU_CST_FA TO VAR1
STORE TRANSFORM(LRU_CST_FA,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO LCF

STORE MTBF_FAC TO VAR1
STORE TRANSFORM(MTBF_FAC,"9999.99") TO VAR2
DO CKLNGTH5
STORE VAR4 TO MF

?' ' + STR(SE_RES_1,4,0) + ' ';
+ SCF1 + '   ';
+ STR(SE_RES_2,4,0) + ' ';
+ SCF2 + ' '
??LRU_ID + ' ';
+ LCF + ' ';
+ MF + ' '
SKIP
ENDDO
?999999
CLOSE DATABASES
RETURN
^Z

**********************************************************
**********************************************************

c                    TPS Max Cost Model
c     B1BTPS.FOR

      DIMENSION TPS(7),YAXIS(2),XAXIS(7,2),XRATIO(7,3)
      DIMENSION ZRATIO(7)
      CHARACTER AA*1
c     need to activate/identify crt/keyboard
      OPEN(5,FILE='USER')
      OPEN(6,FILE='USER')
c     input 7 sample TPS costs from NRLA run.  If more or
```

```
c     less are used all references to (7) or (7, or DO XXXX
c     I=1,7 have to be changed.
      TPS(1)=10.
      TPS(2)=50.
      TPS(3)=100.
      TPS(4)=150.
      TPS(5)=250.
      TPS(6)=500.
      TPS(7)=1000.
c     all NRLA runs were made with 10K hours MTBD or 70K
hours
      YAXIS(1)=10.
      YAXIS(2)=70.
c     the results of the NRLA runs for 7 TPS  values and 2
MTBD's
      XAXIS(1,1)=1.142
      XAXIS(1,2)=7.695
      XAXIS(2,1)=2.07
      XAXIS(2,2)=14.392
      XAXIS(3,1)=3.207
      XAXIS(3,2)=22.622
      XAXIS(4,1)=4.348
      XAXIS(4,2)=30.916
      XAXIS(5,1)=6.685
      XAXIS(5,2)=47.523
      XAXIS(6,1)=12.409
      XAXIS(6,2)=88.902
      XAXIS(7,1)=23.933
      XAXIS(7,2)=171.660
c     compute a ratio between unit costs and MTBD
      DO 100 I=1,7
      XRATIO(I,1)=XAXIS(I,1)/YAXIS(1)
      XRATIO(I,2)=XAXIS(I,2)/YAXIS(2)
  100 CONTINUE
c    print a header to explain what the model does
      WRITE(6,101)
  101 FORMAT(/20X,'B1B TPS MAX COST MODEL'//'  This model
computes',
     &' the MAXIMUM that one should PAY to ACQUIRE an SRU
TPS.'/
     &'  If an SRU TPS cannot be acquired for LESS than this
cost',
     &', then it is more'/'  cost-effective to code the SRU
as a',
     &' DISCARD item.'//4X,'(This model is valid only for the
B1B ',
     &' and should not be used otherwise)'/)
c     return point.
c     ask user for a MTBD
      WRITE(6,201)
  201 FORMAT(//'  MTBD assumptions: 1. Operating Hrs =
Flying',
     &' Hrs.'/20X,'2. QPA = 1.')
  200 WRITE(6,202)
  202 Format(//'  Enter MTBD (000).  Adjust if
necessary.'/'$=')
      READ(*,*)FMTBD
      IF(FMTBD.LT.0.5)GO TO 2000
      IF(FMTBD.GT.150.0)GO TO 2000
      IF(FMTBD.LT.10.)GO TO 206
      IF(FMTBD.GT.70.)GO TO 207
c     we have a stored ratio unit cost/MTBD for 10k and 70k
c     MTBDs. we want to create another ratio for the specific

c     MTBD inputted.
      X=(70.-FMTBD)/60.
  204 DO 205 I=1,7
      XRATIO(I,3)=X*XRATIO(I,1)+(1.-X)*XRATIO(I,2)
  205 CONTINUE
      GO TO 210
  206 X=1.0
      GO TO 204
  207 X=0.0
      GO TO 204
c     enter the unit cost
  210 CONTINUE
      WRITE(6,211)
  211 FORMAT(2X,'Enter the UNIT COST for the SRU
($000)'/'$=')
      READ(*,*)FCOSTS
      IF(FCOSTS.LT.0.5)GO TO 2100
      IF(FCOSTS.GT.1000.)GO TO 2100
c     start out with the ratios based on MTBD
      DO 350 I=1,7
      ZRATIO(I)=XRATIO(I,3)
  350 CONTINUE
c     compute the ratio for user inputted unit cost/MTBD and
c     use it to extrapolate between the seven stored tps
values
      TRATIO=FCOSTS/FMTBD
      IF(TRATIO.LT.ZRATIO(1))GO TO 1700
      IF(TRATIO.GT.ZRATIO(7))GO TO 1800
      DO 1220 I=1,7
      IF(TRATIO.EQ.ZRATIO(I))GO TO 1500
      IF(TRATIO.LT.ZRATIO(I))GO TO 1600
 1220 CONTINUE
 1500 CONTINUE
      WRITE(6,1501)TPS(I)
 1501 FORMAT(//2X,'The MAX TPS COST for this SRU is',
     &'=',F8.1,' ($000)')
      GO TO 2200
 1600 X=ZRATIO(I)-ZRATIO(I-1)
      Y=TRATIO-ZRATIO(I-1)
      Z=TPS(I)-TPS(I-1)
     Z=TPS(I-1)+Z*Y/X
     WRITE(6,1501)Z
     GO TO 2200
1700 WRITE(6,1701)TPS(1)
1701 FORMAT(//2X,'the MAX TPS COST for this SRU is'
    &' less than',F8.1,' ($000)')
     GO TO 2200
1800 WRITE(6,1801)TPS(7)
1801 FORMAT(//2X,'the MAX TPS COST for this SRU is',
    &' greater than',F8.1,' ($000)')
     GO TO 2200
2000 WRITE(6,2001)
2001 FORMAT(//2X,'MTBD IS OUT OF LIMITS OF THE MODEL')
     GO TO 2200
2100 WRITE(6,2101)
2101 FORMAT(//2X,'COSTS ARE OUT OF LIMITS OF THE MODEL')
2200 WRITE(6,2201)
2201 FORMAT(/2X,'DO YOU WISH TO CHANGE PARAMETERS OR RUN',
    &' ANOTHER SRU? (Y/N)'/'$=')
     READ(*,2302)AA
2302 FORMAT(A1)
     IF(AA.EQ.'Y'.OR.AA.EQ.'y')GO TO 200
     IF(AA.EQ.'N'.OR.AA.EQ.'n')GO TO 9900
     WRITE(6,2303)
2303 FORMAT(/2X,'Y OR N ARE ONLY GOOD ANSWERS'//)
     GO TO 2200
9900 CONTINUE
     STOP
     END
^Z
```

Appendix E: Glossary of Terms

AFALC	Air Force Acquisition Logistics Center
AFIT	Air Force Institute of Technology
AFLC	Air Force Logistics Command
AFR	Air Force Regulation
AFSC	Air Force Systems Command
ASD	Aeronautical Systems Division
ATE	automatic test equipment
DBMS	database management system
DOD	Department of Defense
DSS	decision support system
FM	failure mode
FSD	full scale development
IDENT	identifier
ITA	interface test adapter
LCC	life cycle cost
LRU	line replaceable unit
LSA	logistics support analysis
MTBF	mean time between failure
MTBMA	mean time between maintenance action
NRLA	network repair level analysis
OPR	office of primary responsibility
RAM	random access memory
RLA	repair level analysis
SE	support equipment
SPO	system program office

SRU	shop replaceable unit
TIM	technical interchange meeting
TPS	test program set
WPAFB	Wright-Patterson Air Force Base

Appendix F: Definitions of Terms

automatic test equipment: electronic equipment used for the fault detection and repair of complex electronic aircraft avionics components.

database file: the storage location of specific data pertaining to a data record type containing the structure as well as the actual data itself. Database files are distinguished by a .DBF extension.

database management system: the component of a data system with function of storing, retrieving and formatting information to and from the database.

data record: a collection of data stored in data fields within a data record type.

data record type: the categorized group of multiple data records pertaining to a specific class of data (i.e.; weapons system data record type, LRU failure mode data record type, etc.).

depot level maintenance: the highest type of maintenance which includes the complete overhauling, rebuilding, and calibration of equipment as well as the performance of highly complex maintenance actions.

dialog base: the component of a DSS which establishes the degree, format, and method of interface with the user.

DSS group: the group of personnel who augment the DSS development process some of which may use the developed system.

DSS tools: the hardware and software available which is used to directly create a specific DSS.

failure mode: the reason for the component failing (i.e. the subsystem or module that failed within the component).

intermediate level maintenance: the maintenance required, usually performed at base level, to return an end item to its full operational status by the removal of major modules, assemblies, or piece parts.

line replaceable unit: complex electronic aircraft avionics components which can be removed and replaced on an aircraft on the flight line. Also known as "black boxes".

logistics support analysis: an iterative analytical process by which the logistic support necessary of a new system is identified and evaluated.

models base: the component of a DSS which provides a mathematical representation of the structure and relationship between parts of the problem.

repair level analysis: an iterative decision-making process conducted throughout all phases of a system's life cycle with the primary goal of making effective and economical repair level decisions which consider equipment design, operations, and logistics support characteristics.

shop replaceable unit: electronic aircraft avionics components such as modules, assemblies, or printed circuit boards which go into line replaceable units and are removed and replaced in a line replaceable unit in the maintenance shop.

specific DSS: the hardware and software used to support a specific set of decision making tasks.

test program set: the software and hardware used for failure detection and repair of a specific LRU or SRU which controls the testing operations and procedures of the ATE.

user interface: the component of a DSS responsible for providing all interaction between computer and user.

Appendix G: Desk Reference

NRLA INTERFACE

Desk Reference

NRLA INTERFACE version 1.0
Copyright (C) Thomas J. Girz 1988

Table of Contents

NRLA INTERFACE version 1.0

Restricted Rights Warning

NRLA INTERFACE is a copyrighted package designed for the exclusive use of the U.S. Military and is protected by U.S. Copyright Law (Title 17 United States Code). Unauthorized reproduction and/or sales may result in imprisonment of up to ONE YEAR, and fines of up to $10,000 (17 USC 506). Copyright infringers may also be subject to civil liability.

Disclaimer of Warrantee

This software and manual is distributed without any expressed or implied warranties whatsoever. The user is advised to test the program thoroughly before relying on it. The user assumes the entire risk of using this system.

Introduction

This system provides an initial specific decision support system (DSS) that enhances the usefulness of the Network Repair Level Analysis (NRLA) model by integrating a user interface and a database management system (DBMS) that facilitates model manipulation. This system also provides an analysis technique that generates cost guidelines used in the acquisition of test program sets. NRLA INTERFACE was developed for use in conjunction with the NRLA model version 2.0.

This reference guide gives insight into the NRLA INTERFACE system operation. It should be used in conjunction with the Network Repair Level Analysis User's Guide and not in lieu of. All user requirements described in the Network Repair Level Analysis User's Guide pertain when using this system.

Here and subsequently in this reference guide, the term "the system" refers to the applications programs compiled to make up NRLA INTERFACE and this DSS. Any other use of "system" will be specified as such (i.e., weapons system, maintenance system, disk operating system, etc.).

System Requirements

This system can be run on an MS-DOS(*) based PC-compatible micro-computer with a hard drive, a monitor and a 120 column printer using dBase III Plus(*).

Installing the System

In order to install the system, the following steps must be accomplished:

1) The CONFIG.SYS file contained in the directory used to start the computer should contain the commands

```
     FILES = 20
     BUFFERS = 15
```

in order to allow for enough memory for system operation. The CONFIG.SYS file is a text file that can be created with any word processor that creates ASCII text files. If this file is not present, dBase III Plus(*) will operate, but may encounter a "too many files are open" error message. Placing the CONFIG.SYS file on the disk used to start the computer will preclude encountering this error message. Add these lines to the CONFIG.SYS file on your hard drive and reboot the system.

2) Create a separate subdirectory on the hard drive and enter the subdirectory. Install dBase III Plus(*) in this subdirectory as described in the dBase III Plus(*) manual.

3) Place the NRLA INTERFACE disk in drive A: and while in the new subdirectory type <COPY A:*.*>. Keystroke entries required by the user are here and subsequently indicated by their inclusion in pointed brackets (<...>).

4) Place the NRLA version 2.0 disk in drive A: and type <COPY A:NRLA*.*>.

These steps will install dBase III Plus(*), the 68 applications programs and 13 database files that make up NRLA INTERFACE, the NRLA model, and the TPS Max Cost model.

Starting the System

In order to start NRLA INTERFACE, the user must enter the subdirectory containing the above mentioned programs, start the dBase III Plus'®' system, and type <DO NRLAFACE> at the dot prompt.

The system displays the Rights and Warranties screen outlining the copyrights and warranties associated with using this software package. Pressing any key will bring up the Main Menu.

Main Menu

The Main Menu is the highest level of operation in this system. In order to enact the operations of the Display, Edit, Create, or Run Analysis modules, the user need only enter the highlighted letter of his selection. The user must ensure that the Caps Lock key is engaged in order for the system to recognize a correct keystroke entry.

At the Main Menu the user has a choice of the following keystroke entries: <D>, <E>, <C>, <R>, or <X>. With these entries the user can display, edit, or create data files, run analysis models or exit the system to the computer's disk operating system. The system will only accept the above responses at the Main Menu; any other entry will not be recognized. This screen will continue to be displayed until one of these selections is entered.

Displaying Data

With a selection of <D> at the Main Menu selection prompt the system moves into the display module. At the Data Display Screen the user can enter the highlighted selection associated with any of the 11 data record types to display the data contained within each. The user can only display the data through this module; no interaction with the database file is possible. This module provides the user a means to quickly display the data presently stored in the database. It allows the user to verify the correctness of entered data quickly without having to specify individual records to view.

At the Data Display Screen selection prompt the system will only accept the following entries: <W>, <M>, <U>, <E>, <L>, <F>, <S>, <C>, <O>, <H>, <I>, <A>, or <R>. Any other entry will not be recognized and the Data Display Screen will remain on the screen until one of these selections is entered.

Upon entry of a highlighted selection pertaining to a data record type, the system displays the data presently stored in the database file for the data record type selected. A selection of <A> displays all the data stored in each of the data record type database files without having to select individual data record types. A selection of <R> returns the user to the Main Menu.

Upon viewing each data record the user has the option of remaining in the Display Module or exiting and returning to the Main Menu. The default to the "Exit Display Module (Y or N)?" prompt is "N". An entry of <N> displays the next record in the file or returns to the Data Display Screen if at the end of the database file. An entry of <Y> returns the system to the Main Menu.

If a data record type is selected that contains no data, the system will inform the user that no data is stored in this data record type.

Editing Data

A selection of <E> at the Main Menu selection prompt displays the Data Edit Screen. At this screen the user can enter the highlighted selection associated with any of the 11 data record types to edit the data contained within each.

In order to edit data contained in the weapons system, maintenance system, supply system, output options, or wholesale factors data record types, the user need only enter the highlighted character associated with each of these data record types. These data record types contain only one data record each. Upon the entry of <W>, <M>, <U>, <O>, or <H> the system moves directly to the edit data record screen and allows interaction with the database for editing. Should the user choose not to edit data contained in the data record type selected, he may step through the fields on the screen by pressing the <return> key. Pressing <pg dn> will exit the data record edit screen without changing further any data within the database file not previously edited. Care should be taken in editing data because data edited prior to pressing <pg dn> will be saved in the database file as edited upon pressing <pg dn>.

In order to edit data contained in the support equipment, LRU, LRU failure mode, SRU, LRU/SRU/SE cross reference, or individual factors data record types, the user would enter the appropriate highlighted character. Upon selection of <E>, <L>, <F>, <S>, <C>, or <I> the system displays an Edit Choice Menu. This enables the user to add, delete or edit any of the multiple data records within each of these data record types.

The user can enter the highlighted selection associated with the operation desired. The system will only accept entries of <A>, <D>, <E>, or <R>. Any other entry will not be recognized.

Adding Data Records. An entry of <A> at the Edit Choice Menu Screen enables the user to add a new data record within the specific data record type. The add data record screens perform in an interactive mode to receive data and add a data record to the database file. Following each data record addition, the system asks if the user wishes to enter another data record. Upon entry of <Y>, the data previously entered are saved and the fields are cleared. A new data record may now be entered. Upon an entry of <N> the system will return to the Edit Choice Menu.

Blank records cannot be entered in the database files. In order to keep the user from entering a blank data record, the system checks a significant data field within the data record types. The significant data fields for each data record type are listed in Table I.

Table I. Significant Data Fields

DATA RECORD TYPE	SIGNIFICANT DATA FIELD
Support Equipment Data	SE Resource Number (SERN)
LRU Data	LRU Identifier
LRU Failure Mode Data	LRU Identifier
SRU Data	SRU Identifier
LRU/SRU/SE Cross Reference Data	LRU or SRU Identifier
Individual Factors	LRU Identifier

If the user enters a blank field into one of the significant data fields associated with the data record types shown in Table I, the system assumes that the user is trying to enter a blank record. The system will notify the user that the entry of a blank record has been attempted and offers the option to abort the addition of this blank data record. The system defaults to <Y> at the Abort New Record Addition prompt and the user must enter an <N> in order to save this data record addition. Following this, the user has the option to enter another record.

Deleting Data Records. An entry of <D> at the Edit Choice Menu enables the user to delete data records within the specific data record type. The system will list the data records stored in the database file for the record type selected at the Data Edit Screen. The data records are listed in ascending order of the significant data field. The significant data fields are the same as those listed in Table I.

Record numbers may or may not be in order depending on the order in which the data was originally entered. Record numbers show the order that the records are stored in within the database file.

Due to screen size limitations, only 15 data records are listed at a time. The user has the option of listing the next 15 records with the default to the Show Next 15 Records prompt being <Y>. A response of <N> by the user ends the data records listing and the system brings up a screen which allows the user to enter the value of the significant field for the data record be to deleted.

If no deletion is desired, the user may enter <9999> and the system will return to the Edit Menu with no data record being deleted from the database file.

Upon entry of an invalid significant field, that is, one that is not stored in the database file of the specified record type, the system informs the user that there is no such data record containing the significant field entered. The system then returns to the Edit Menu upon pressing any key.

Upon entry of a valid significant field, that is, one that is stored in the database file of the specific record type, the user is required to confirm data record deletion. The default at this deletion confirmation prompt is <N>. An entry of <Y> will delete the data record containing the significant field shown and the system returns to the Edit Choice Menu for the record type chosen at the Data Edit Screen.

The data record deletion process is the same for each of the data record types listed in Table I. The only exceptions are the LRU Failure Mode and LRU/SRU/SE Cross Reference Data Record Types.

Because of the nature of the data stored in the database files for the LRU Failure Mode and LRU/SRU/SE Cross Reference Data Record Types, multiple data records stored in the database file may contain the same value for the significant field. A second discriminator, therefore, distinguishes between data records with identical values in the significant fields in order to be able to identify the correct data record to delete or edit. For both the LRU Failure Mode Data and the LRU/SRU/SE Cross Reference Data Record Types, this second discriminator is the Failure Mode Number. When deleting data records from these data record types the user must enter the significant field to be deleted. Following this entry, the user must enter the value of the Failure Mode Number in order to distinguish between LRU ID Numbers.

Editing Data Records. An entry of <E> at the Edit Choice Menu enables the user to edit data records within the specific data record type. The system will list the data records stored in the database file for that record type. Similar to deleting data, the user can list the data records by significant field and enter the significant field of the data record he wants to edit.

The system performs the same check for a valid significant field as when deleting data records. Blank records cannot be entered in the database files. In order to keep the user from entering a blank data record, the system checks a significant data field within the data record types. The significant data fields for each data record type are listed in Table I.

When a valid significant field is entered, the system displays an interactive screen that allows the user to change data in any of the fields within the data record. In order to edit, the user can press either <return> or <down arrow> to place the cursor on the field to edit. Upon entering <return> after editing a field, the edited data is stored in the database file.

When the edit is complete the system returns to the Edit Choice Menu. Should the user choose not to edit data contained in these records, he may step through the fields on the screen by pressing the <return> key. Pressing <pg dn> will exit the data record screen without any further changes to the data in the database file not previously edited. The system then returns to the Edit Choice Menu.

The Edit Data Records Screens work in a similar fashion for each of the data record types listed in Table I. As with the Delete Data Records Screens, the only exceptions are the LRU Failure Mode Data and the LRU/SRU/SE Cross Reference Data Record Types. The same process to distinguish between data records with identical values in the significant fields using a second discriminator is followed.

Creating a New File

A selection of <C> at the Main Menu selection prompt allows the user to create new data records. In this module the user can create new database files containing new data for any or all data record types. In order to do so the system deletes the data currently in the database file for the record type selected and allows the user to enter data records into a blank database file. It essentially wipes the slate clean in order to enter data for a different weapons system.

To ensure that the user desires to create a new database file for the record type selected, a warning and confirmation screen is displayed and the user must confirm continuation of the creation module.

If the user does not want to erase and create a new database file for the selected record type, a response of <R> will return the system to the Main Menu. With a selection of <C> to continue the creation module, the system will delete the records stored in the database files for the specified data record type and will display the same screens as those displayed for adding data records in order to create/enter new data records into the database files. Similarly the system will not allow blank records to be entered.

When creating a new file, the Network Repair Level Analysis User's Guide should be consulted regarding data requirements.

Running the Analysis Models

A selection of <R> at the Main Menu selection prompt displays the Run Analysis Menu. This module enables the user to run the analysis models portion of the system.

At this menu the user has the option to run either the NRLA model or the TPS Max Cost model. The user may also display the directory of .DAT files stored on the hard drive or return to the Main Menu. At this prompt the system only accepts entries <N>, <T>, <D>, or <R>. Any other entry is not recognized by the system.

Running the NRLA Model. In order to run the NRLA model, the user selects <N> at the Run Analysis Menu selection prompt and the system prompts the user to enter the date. Following entry of the date, the user must enter the filenames to which the data stored in the database files will be written. NRLA will use the data written to these files for its analysis. Any filename up to eight characters long with a .DAT extension may be entered. The .DAT extension is required for the NRLA model to process the data and must be entered as part of both filenames. Upon entry of the filenames the system writes the data to the specified files and, as the data is being written, displays the screen showing the filenames to which the data is being written.

At this point the NRLA model is enacted and the user should press the <control> and <P> keys simultaneously in order to send the output to the printer. If no printout is desired, the user enters <OK> to continue the NRLA model. A blank diskette need not be inserted in the default as specified. The user need only enter the filenames for the Run Specific Data and the Program Specific Data including the .DAT extension. Upon entry of the data filenames, the NRLA model will run.

To abort the NRLA analysis, the user can press the <control> and <C> keys simultaneously. The analysis will stop and the system will return to the Main Menu.

Following completion of the printing of the NRLA analysis the user should again simultaneously press <control> and <P> to disengage the printer.

Running the TPS Max Cost Model. A selection of <T> at the Run Analysis Menu selection prompt will run the TPS Max Cost model. This model requires that the LRU/SRU mean time between demand (MTBD) as well as the unit cost of the LRU/SRU under analysis be entered. The model will then return the maximum cost that the government should pay in acquiring a TPS. A printout of this analysis may be obtained by simultaneously pressing <control> and <P> prior to entering <T> at the Run Analysis Menu. After the printing of the analysis results this procedure should be undertaken to disengage the printer.

Viewing .DAT Files. An entry of <D> at the Run Analysis Menu selection prompt allows the user to view the list of filenames that are saved on the hard drive with the .DAT extension. This permits the user to see what has been saved with a .DAT extension should he desire to write data to files with different filenames.

Files created previous to the development of NRLA INTERFACE can be stored in this subdirectory with a .DAT extension and can be accessed using this system. By naming the filename with a .DAT extension, the user can view easily what has been saved. The user may run the NRLA model using .DAT files not created through NRLA INTERFACE. These files cannot, however, be edited. In order to do so, the data must be entered through NRLA INTERFACE and stored separately. The procedure for storing data separately is explained in the next section.

Saving Data for Different Weapons Systems

Should the user desire to use NRLA INTERFACE and this DSS to create, edit, and store data for several weapons systems, the data for each weapons system should be stored in separate subdirectories on a five and one-quarter inch floppy diskette.

The 13 database files stored in the subdirectory used here contain the structure of the data as well as the actual data pertaining to the weapons system under analysis. When performing analysis on several weapons systems, these 13 database files should be carefully stored so as not to write data for one weapons system over the data of another weapons system.

For example, if the user has been running the NRLA model for Weapons System A, the database files on the hard drive contain all the relevant data for that weapons system. If the user then desires to run the NRLA model for Weapons System B and not lose the data for Weapons System A, he must follow these steps:

1) At the NRLA INTERFACE Main Menu, enter <X> to exit to the computer's disk operating system.

2) Insert a blank diskette into drive A:.

3) Make a subdirectory on the drive A: diskette for storing the database files containing Weapons Systems A data.

4) Enter this drive A: subdirectory and type <COPY C: *.DBF A:>

Each of the 13 database files will be copied to this drive A: subdirectory. The user may now enter the DSS and run the create module to store data for Weapons System B. The above steps should be followed to save Weapons System B data to a floppy diskette making sure the database files (i.e., those with .DBF extensions) are stored in a different subdirectory than those stored for Weapons System A.

Once the database files for Weapons System B are stored on a floppy diskette, the data for Weapon System A can be copied back to the hard drive for further data manipulation and analysis. This is done from the disk operating system by entering the correct directory on the drive A: diskette and typing <COPY A: *.DBF C:>. This will overwrite the 13 .DBF files on the hard drive with those contained on the drive A: diskette. This data may now be manipulated through NRLA INTERFACE.

Data Entry Guidelines

As stated previously, the data requirements covered in the Network Repair Level Analysis User's Guide still pertain when using this system for data and model manipulation. However, due to certain programming constraints the following guidelines are provided in order to effectively use this system:

1) For the Weapons System Data Records Type,

a) the value entered for Ratio Force Overseas must be less than 1.0.

b) the value entered for the Lower Sensitivity Range must be less than .99.

c) the value entered for the Sensitivity Type must be 1 or 0.

2) For the Maintenance System, Supply System, Support Equipment, LRU, LRU/SRU/SE Cross Reference, Output Options, Wholesale Factors, and Individual Factors follow the guidelines in the Network Repair Level Analysis User's Guide.

3) For the LRU Failure Mode data record type the following values will be accepted in the Forced LRU Failure Mode Decisions field:

a) a value of 100 excludes Depot Maintenance for the failure mode;

b) a value of 010 excludes Scrap for the failure mode;

c) a value of 001 excludes Intermediate Maintenance for the failure mode;

d) a value of 110 excludes Depot Maintenance and Scrap;

e) a value of 101 excludes Depot Maintenance and Intermediate Maintenance;

f) a value of 011 excludes Scrap and Intermediate Maintenance;

g) a value of 000 excludes none;

h) a value of 111 is considered an error. The NRLA model will change the 1's to 0's and the program will proceed as if no exclusions are desired for the particular item.

4) For the SRU Data Records Type, the same values are valid for the Forced SRU Decisions field as those for the Forced LRU Failure Mode Decisions field.

Bibliography

1. Air Force Acquisition Logistics Center. Abstracts of Lessons Learned. Lesson Number 0272. Wright-Patterson AFB OH: 1 January 1987.

2. Air Force Acquisition Logistics Center. Abstracts of Lessons Learned. Lesson Number 0458. Wright-Patterson AFB OH: 1 January 1987.

3. Air Force Acquisition Logistics Center. Network Repair Level Analysis Model User's Guide. Wright-Patterson AFB OH: January 1986.

4. Air Force Acquisition Logistics Center. The Network Repair Level Analysis Instant Expert Guide. Wright-Patterson Air Force Base OH: AFALC. 2 June 1986.

5. Air Force Acquisition Logistics Center. The Network Repair Level Analysis Programmer's Guide. Wright-Patterson AFB OH: AFALC/LSS. January 1986.

6. Allen, Mary K. and Margaret A. Emmelhainz, "Decision Support Systems: An Innovative Aid to Managers," Journal of Business Logistics, Vol 5, No 2: 128-142, 1984.

7. Bodnar, Albert F. A Simulation Model to Evaluate Multiyear Procurement Economics For Spares Acquisition. MS Thesis, AFIT/GSM/LSP/85S-2. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1985.

8. Callahan, James. Automatic Test Equipment Manager, B2 System Program Office. Personal interviews. AFALC/YSL, Wright-Patterson AFB OH, 10 September 1987 through 21 May 1988.

9. Davis, Capt Carl L. Class lecture in COMM 630, Research Methods. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 29 February 1988.

10. Davis, Michael W. Applied Decision Support. Englewood Cliffs NJ: Prentice-Hall, Inc., 1988.

11. Department of Defense. Acquisition and Management of Integrated Logistics Support for Systems and Equipment. DOD Directive 5000.39. Washington DC: General Printing Office.

12. Department of Defense. Defense Acquisition Program Procedures. DOD Instruction 5000.2. Washington DC: General Printing Office, 1 September 1987.

13. Department of Defense. Logistics Support Analysis. Military Standard 1388-1a. Washington DC: Government Printing Office. 11 April 1983.

14. Department of the Air Force. Acquisition of Support Equipment. Air Force Regulation 800-12, Attachment 1. Washington DC: HQ USAF, 13 December 1985.

15. Department of the Air Force. Repair Level Analysis (RLA) Procedures. AFLC/AFSC Pamphlet 800-4. Wright-Patterson AFB OH: HQ AFLC and Washington DC: HQ AFSC, 25 November 1983.

16. Department of the Air Force. Repair Level Analysis (RLA) Program. AFSC/AFLC Regulation 800-28. Washington DC: HQ AFSC and Wright-Patterson AFB OH: HQ AFLC, 29 May 1981.

17. Douglas, Robert A. Operations Research Analyst, B-2 System Program Office. Personal interviews. AFALC/YSL, Wright-Patterson AFB OH, 5 January through 22 August 1988.

18. Emmelhainz, Maj Larry W. Class lectures in LOGM 615, Decision Support Systems. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 28 March - 3 June 1988.

19. Emory, C. William. Business Research Methods. Homewood IL: Richard D, Irwin, Inc., 1985.

20. Fisher, Kevin R. Systems/Programmer Analyst. Telephone interviews. Cumberland Advisors, Vineland NJ, 15 March 1988 and 25 August 1988.

21. Girz, Thomas J. and Calvin J. Romrell, Development of a Decision Support System Using the Network Repair Level Analysis Model for Determining Test Program Set Acquisition Costs. For LOGM 615, Decision Support Systems. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 16 May 1988.

22. Jones, Edward. dBase III Plus Programmer's Reference Guide, Indianapolis IN: Howard W. Sams and Company, 1987.

23. Pangborn, Linda, NRLA Analyst. Personal interviews. AFALC/LSX, Wright-Patterson AFB OH, 23 November 1987 through 11 March 1988.

24. Sprague, Ralph H. and Eric D. Carlson. Building Effective Decision Support Systems. Englewood Cliffs NJ: Prentice-Hall, Inc., 1982.

25. Sprague, Ralph H. and Hugh J. Watson. Decision Support Systems. Englewood Cliffs NJ: Prentice-Hall, Inc., 1986.

26. Stultz, Russell A. The Illustrated dBase III Plus Book. Dallas: Wordware Publishing, Inc., 1987.

Vita

Thomas J. Girz was born on [redacted] He graduated from Lorain Catholic High School in 1979 and received a Bachelor of Science degree in Industrial and Systems Engineering from Ohio University in 1983. His career with the USAF began as an Industrial Engineer working for the DCS/Manpower and Personnel at Air Force Logistics Command Headquarters at Wright-Patterson AFB, Ohio. Duties included performing manpower studies to determine manning requirements at HQ AFLC. In November of 1984, he took a job as an operations research analyst in the Air Force Acquisition Logistics Center working in the Plans and Analysis Section of the Logistics Directorate in the B-1B System Program Office (SPO). Duties there included performing special studies to determine logistics requirements for avionics test equipment and management of interim contractor support contracts. In October 1986, he moved to the Defensive Avionics Directorate of the B-1B SPO where he managed electronic countermeasures retrofit activities. In June 1987, he entered the School of Systems and Logistics at the Air Force Institute of Technology.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GLM/LSM/88S-24

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Systems and Logistics

6b. OFFICE SYMBOL *(If applicable)*
AFIT/LSM

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology (AU)
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Development of a Decision Support System Using Network Repair Level Analysis for Determining Test Program Set Acquisition Costs

12. PERSONAL AUTHOR(S)
Thomas J. Girz, B.S.

13a. TYPE OF REPORT
M.S. Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1988 September

15. PAGE COUNT
222

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
15	05	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Database, repair level analysis, maintenance planning, computer applications, decision support system, acquisition management, test program set, user interface

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Thesis advisor: Charles F. Youther
Assistant Professor of Logistics Management

Approved for public release IAW AFR 190-1.

WILLIAM A. MAUER 17 Oct 88
Associate Dean
School of Systems and Logistics
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Charles F. Youther

22b. TELEPHONE *(Include Area Code)*
(513)255-5023

22c. OFFICE SYMBOL
AFIT/LSM

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

The purpose of this research was to determine the applicability of developing a software system to enhance as well as expand the usefulness of the Network Repair Level Analysis (NRLA) model. Maintenance planners and acquisition managers need new tools in order to effectively plan and manage support programs in the face of increasing budget cuts. This research provides an initial specific decision support system (DSS) that enhances the usefulness of the NRLA model by providing a user interface and a database management system (DBMS) which facilitates model manipulation. This initial specific DSS also expands the model's usefulness by providing an analysis technique that generates cost guidelines used in the acquisition of test program sets.

Management personnel from the Aeronautical Systems Division at Wright-Patterson Air Force Base were interviewed and consulted throughout the system development process. The DSS was developed using dBase III Plus(R). This included writing applications programs that make up the user interface and the DBMS, and integrating them with a database and the analysis models.

This DSS can be installed and used on an IBM(R) compatible microcomputer with a 20 megabyte hard drive, a monitor, and a 120 column printer.